On the Axiomatisability of Priority III: Priority Strikes Again

Luca Aceto[a,b], Elli Anastasiadi[b], Valentina Castiglioni[b,1], Anna Ingólfsdóttir[b], Bas Luttik[c], Mathias Ruggaard Pedersen[b]

[a] *Gran Sasso Science Institute, L'Aquila, Italy*
[b] *Reykjavik University, Reykjavik, Iceland*
[c] *Technische Universiteit Eindhoven, Eindhoven, The Netherlands*

Abstract

Aceto et al., proved that, over the process algebra BCCSP with the *priority operator* of Baeten, Bergstra and Klop, the equational theory of *order-insensitive bisimilarity* is not finitely based. However, it was noticed that by substituting the action prefixing operator of BCCSP with BPA's *sequential composition*, the infinite family of equations used to show that non-finite axiomatisability result could be proved by a finite collection of sound equations. That observation left as an *open question* the existence of a finite axiomatisation for order-insensitive bisimilarity over BPA with the priority operator. In this paper we provide a negative answer to this question. We prove that, in the presence of at least two actions, order-insensitive bisimilarity *is not finitely based* over BPA with priority.

Keywords: Finite Axiomatisations, Bisimilarity, Priority Operator, Sequential Composition

1. Introduction

Process algebras [7, 12] are a classic tool for reasoning about the behaviour of concurrent and distributed systems, or *processes*. Briefly, the *operational semantics* [27] of a process is modelled via a *Labelled Transition System* (LTS) [19] in which the computational steps are abstracted into state-to-state transitions having *actions* as labels. Then, behavioural equivalences, like *bisimulation equivalence* [26], are defined on the LTS in order to compare the behaviour of processes. This comparison is crucial for system *verification*: to verify that the actual system meets its specification we check whether their LTSs are behaviourally equivalent. To this end, an *equational axiomatisation* of the behavioural equivalence of interest is provided, as it allows for proving valid equations over processes by *replacing equals by equals*.

A fundamental feature that has been implemented within the process algebra framework is the possibility to express that some actions have *priority* over others (we refer the interested reader to [15] for an overview of the proposals). This allows for modelling, for example, that an interrupt or shutdown action may be needed when a system deadlocks or starts exhibiting erroneous behaviour, and, likewise, that a scheduler needs to assign a different level of urgency to actions based on its scheduling policy. Here we consider the approach taken in [8], where a priority operator Θ is introduced. This operator is based on an irreflexive partial order, called the *priority order*, over the actions that are available to the process, and only allows an action to be performed if no other action with a higher priority is possible at the given moment.

In the literature we can find a variety of results on the equational theory of the priority operator Θ in different settings, as we review below. With this paper, we give our contribution to these studies by discussing the equational axiomatization for a process algebra having both Θ and the *sequential composition operator* of BPA [11], modulo a notion of bisimulation equivalence, called *order-insensitive bisimilarity* [3], that holds *irrespectively* of the chosen priority order over actions.

[1]Corresponding author: email addresses *valentinac@ru.is, vale.castiglioni@gmail.com*

1.1. *On the axiomatisability of priority*

Earlier studies on the axiomatisability of the priority operator were carried out with respect to a chosen, arbitrary, priority order. In the seminal papers [8, 10] it was shown that, provided that the set of actions is finite, the priority operator admits a *finite*, *ground-complete* equational axiomatisation. (A set of axioms is called *ground-complete* if every sound equation between process terms without variables can be derived from those axioms using the rules of equational logic.) For an infinite set of actions, it was proved in [2] that the operator Θ admits no finite equational axiomatisation over the process algebra BCCSP_Θ, which consists of basic operators from CCS [21] and CSP [18], enriched with Θ. Furthermore, a specific priority order was exhibited for which no finite equational ground-complete axiomatisation exists.

Later, in [3], the first study of an equational axiomatisation of an equivalence that is irrespective of the chosen priority order was provided. More precisely, it considers the notion of *order-insensitive bisimilarity*, denoted by $\underline{\leftrightarrow}_*$, over processes in BCCSP_Θ: two processes are $\underline{\leftrightarrow}_*$-equivalent if they are bisimilar under every priority order. Now, one may expect that if we consider order-insensitive bisimilarity then there are no sound equations of interest that involve the priority operator. However, as shown in [3], this is not the case. If the set of actions contains at least two distinct elements, then there is no finite, ground-complete equational axiomatisation modulo order-insensitive bisimilarity. To prove their negative result, the authors of [3] showed that no finite set of equations valid modulo $\underline{\leftrightarrow}_*$ can prove all of the equations in the following infinite family

$$a^n.(b+c) + a^n.b + a^n.c \;\approx\; a^n.(b+c) + a^n.b + a^n.c + a^n.\Theta(b+c) \qquad (n \geq 0) \;. \tag{E}$$

However, they also remarked that if we replace BCCSP's action prefixing with BPA's *sequential composition* operator, then all the equations in (E) could be replaced by the following valid equation

$$x \cdot (b+c) + x \cdot b + x \cdot c \;\approx\; x \cdot (b+c) + x \cdot b + x \cdot c + x \cdot \Theta(b+c) \;.$$

This observation left the following open problem:

Is order-insensitive bisimilarity finitely axiomatisable over the process algebra BPA_Θ, namely BPA enriched with the priority operator? (P)

In this paper, we provide a *negative* answer to this question.

1.2. *Our contribution*

Our main result consists in proving that, provided there are at least two distinct actions, the priority operator admits no finite, ground-complete equational axiomatisation modulo order-insensitive bisimilarity over the process algebra BPA_Θ.

The first issue we need to overcome is that, differently from classical bisimulations, order-insensitive bisimilarity is not coinductive: the derivatives of two order-insensitive bisimilar processes cannot be, in general, paired-up in order-insensitive bisimilarity equivalence classes. Hence, we will first of all identify a class of processes on which order-insensitive bisimilarity always behaves coinductively (Proposition 3).

Then, to prove our negative result we use proof-theoretic techniques that have their roots in Moller's classic results to the effect that bisimilarity is not finitely based over CCS (see, e.g., [4, 22–24]). Roughly speaking, we will identify a special property of processes, called the (n,Θ)-*dependency* property, associated with each finite set $\mathcal{E}$ of sound axioms and a natural number n. Informally, a process satisfies (n,Θ)-dependency if by performing a trace of length n it reaches a process whose behaviour depends on the considered priority order, and is thus determined by the priority operator. Moreover, we require that, at each step, the process has the possibility of terminating. The idea is that, when n is *large enough*, whenever an equation $p \approx q$ is derivable from $\mathcal{E}$, then either both terms p and q satisfy (n,Θ)-dependency, or none of them does. The negative result is then obtained by exhibiting an infinite family of valid equations $\{e_n \mid n \geq 0\}$ in which the (n,Θ)-dependency property is not preserved, that is, for each $n \geq 0$, only one side of e_n satisfies (n,Θ)-dependency. Due to the choice of the special property, this means that the equations in the family cannot all be derived from a finite set of valid axioms and therefore no finite, sound axiom system can be complete (Theorem 1). We remark that the requirement on the possibility of termination after each step will ensure that the processes on both sides of the equations e_n cannot be written as a sequential composition, thus preventing the replacement of the infinite family with a finite number of equations that occurred in the case of the equations in (E).

In the axiom system ACP_Θ the axioms for the priority operator made use of an auxiliary operator, called the *unless* operator. It is then natural to wonder whether by adding also the unless operator to the syntax of BPA_Θ it would be possible to obtain a finitely based axiomatisation of order-insensitive bisimilarity. We show that also in this case the answer is negative (Theorem 4).

Finally we study the complexity of the order-insensitive bisimilarity checking. As two processes are order-insensitive bisimilar if and only if they are bisimilar under all possible priority orders, the simplest algorithm for order-insensitive bisimilarity would consists in checking all of them. Our main contribution to this problem is not in the cost of a bisimilarity check, which can be done in $O(m_t \log m_s)$, where m_t is the number of transitions and m_s the number of states [25], but it consists in showing that we actually need to do the check for all possible priority orders. In fact, we prove that for each priority order there exists at least a pair of processes that are bisimilar with respect to all priority orders with the sole exception of the chosen one (Theorem 6). Following [20], there are $2^{k^2/4+3k/4+O(\log k)}$ partial orders over a set of k actions. Hence, we show that the problem of deciding whether two processes are order-insensitive bisimilar is in **coNP** and can be solved in time $2^{k^2/4+3k/4+O(\log k)} \cdot O(n^2)$, where n is the sum of the sizes of the two processes (Theorem 5).

1.3. Outline of the paper

We start by reviewing background notions in Section 2. Section 3 gives an informal presentation of our proof strategy, whose technical development is provided in Sections 4–7. In detail: Section 4 comes with technical results necessary to reason on the semantics of open process terms. In Section 5 we provide the properties necessary to ensure that order-insensitive bisimilarity behaves coinductively. In Section 6 we present the (n, Θ)-dependency property of processes necessary to prove our negative result. Our main result is in Section 7 where we prove that the order-insensitive bisimilarity is not finitely based over BPA with the priority operator. In Section 8 we briefly argue that the negative result would still hold even if we enrich the syntax of BPA_Θ with the auxiliary operator unless. Then, we devote Section 9 to discussing the complexity of order-insensitive bisimilarity checking. Finally, we draw some conclusions and discuss future work in Section 10.

1.4. What's new

A preliminary version of this paper appeared as [1]. Besides providing the full proofs of our results and new examples, we have enriched our previous contribution as follows:

a. We discuss the general reasoning behind the proof of our main result (Theorem 1) and present our proof strategy at an informal level, thus providing a guide for the reader through the technical development of our result (Section 3).

b. We discuss the possibility of using auxiliary operators to axiomatise the priority operator Θ and thus regaining a finite ground-complete axiomatisation over the enriched language BPA_Θ, modulo bisimilarity. We argue that due to some features of order-insensitive bisimilarity, this is not the case (Section 8).

c. We discuss the complexity of order-insensitive bisimilarity check and we show that it is indeed necessary to always check for bisimilarity with respect to all priority orders (Section 9).

2. Background

In this section we review some preliminary notions on operational semantics and equational logic. Since our work naturally builds on [3, 5] we will use the notation from those papers as much as possible.

$$(r_1)\ \frac{}{a \xrightarrow{a}_{>} \surd} \qquad (r_2)\ \frac{p \xrightarrow{a}_{>} \surd}{p \cdot q \xrightarrow{a}_{>} q} \qquad (r_3)\ \frac{p \xrightarrow{a}_{>} p'}{p \cdot q \xrightarrow{a}_{>} p' \cdot q}$$

$$(r_4)\ \frac{p \xrightarrow{a}_{>} \surd}{p + q \xrightarrow{a}_{>} \surd} \qquad (r_5)\ \frac{q \xrightarrow{a}_{>} \surd}{p + q \xrightarrow{a}_{>} \surd} \qquad (r_6)\ \frac{p \xrightarrow{a}_{>} p'}{p + q \xrightarrow{a}_{>} p'} \qquad (r_7)\ \frac{q \xrightarrow{a}_{>} q'}{p + q \xrightarrow{a}_{>} q'}$$

$$(r_8)\ \frac{p \xrightarrow{a}_{>} \surd \quad \forall\, b > a \,.\, p \stackrel{b}{\not\rightarrow}_{>}}{\Theta(p) \xrightarrow{a}_{>} \surd} \qquad (r_9)\ \frac{p \xrightarrow{a}_{>} p' \quad \forall\, b > a \,.\, p \stackrel{b}{\not\rightarrow}_{>}}{\Theta(p) \xrightarrow{a}_{>} \Theta(p')}$$

Table 1: Operational semantics of processes in BPA_Θ.

2.1. BPA_Θ: syntax and semantics

The syntax of *process terms* in BPA_Θ, namely BPA [11] enriched with the priority operator [8], is generated by the following grammar

$$t ::= a \mid x \mid t \cdot t \mid t + t \mid \Theta(t) \ ,$$

with a ranging over a set of actions $\mathcal{A}$, x ranging over a countably infinite set of variables $\mathcal{V}$ and t ranging over process terms. We write $\mathrm{var}(t)$ for the set of variables occurring in t. A process term is *closed* if no variable occurs in it. We shall, sometimes, refer to closed process terms simply as *processes*. We let $\mathbf{P}$ denote the set of BPA_Θ processes and let $p, q, \ldots$ range over it.

We use the *Structural Operational Semantics* (SOS) framework [27] to equip processes with a semantics. A *literal*, or *open transition*, is an expression of the form $t \xrightarrow{a} t'$ for some process terms t, t' and action $a \in \mathcal{A}$. It is *closed* if both t, t' are closed process terms.

The inference rules for *sequential composition* $\cdot$, alternative *nondeterministic choice* $+$ and *priority* Θ are reported in Table 1. We remark that the semantics of Θ is based on a strict irreflexive partial order $>$ on $\mathcal{A}$, called the *priority order*, which justifies the parametrization of the derived transition relation with respect to $>$. For simplicity, given $a, b \in \mathcal{A}$, we write $a > b$ for $(a, b) \in >$. To deal with sequential composition in the absence of deadlock and empty process (see, e.g., [11, 29]), we introduce the *termination predicate* $\rightarrow_{>} \surd \subseteq \mathbf{P} \times \mathcal{A}$. Intuitively, $t \xrightarrow{a}_{>} \surd$ means that t can terminate successfully in one step by performing action a.

A *substitution* σ is a mapping from variables to process terms. It extends to process terms, literals and rules in the usual way and it is *closed* if it maps every variable to a process. We denote by $\sigma[x \mapsto u]$ the substitution that maps each occurrence of the variable x into the process term u and behaves like σ over all other variables.

In [6] it was shown that we can define a stratification [14, 17] on the set of BPA_Θ rules by counting the number of occurrences of the priority operator in the left-hand side of a transition. Hence, the inference rules in Table 1 induce a unique supported model [6, 16] corresponding to the $\mathcal{A}$-labelled transition system $(\mathbf{P}, \mathcal{A}, \rightarrow_{>}, \rightarrow_{>} \surd)$ whose transition relation $\rightarrow_{>}$ (respectively, predicate $\rightarrow_{>} \surd$) contains exactly the closed literals (respectively, predicates) that can be derived by structural induction over processes using the rules in Table 1.

As usual, we write $p \xrightarrow{a}_{>} p'$ for $(p, a, p') \in \rightarrow_{>}$, $p \rightarrow_{>} p'$ if $p \xrightarrow{a}_{>} p'$ for some $a \in \mathcal{A}$, and $p \stackrel{a}{\not\rightarrow}_{>}$ if there is no p' such that $p \xrightarrow{a}_{>} p'$. For $k \in \mathbb{N}$, we write $p \rightarrow_{>}^{k} p'$ if there are $p_0, \ldots, p_k$ such that $p = p_0 \rightarrow_{>} \cdots \rightarrow_{>} p_k = p'$. Furthermore, for a sequence of actions $s = a_1 \ldots a_n$, we write $p \xrightarrow{s}_{>} p'$ to mean that $p \xrightarrow{a_1}_{>} p_1 \xrightarrow{a_2}_{>} \cdots p_{n-1} \xrightarrow{a_n}_{>} p'$ for some processes $p_1, \ldots, p_{n-1}$.

We associate two classic notions with each process: its *depth* and its *norm*. As usual, they express, respectively, the length of a *longest* and a *shortest* sequence of transitions that are enabled for the process. Since in our setting the length of sequences of enabled transitions depends on the considered priority order, we define the depth and the norm of a process with respect to the empty order. The reason for this choice is twofold. Firstly, we notice that the depth defined with respect to the empty order is an upper bound for the depths defined with respect to any other priority order. Since for our purposes we will need to consider upper bounds for the depth of processes, and not the exact value of their depths, it is reasonable to consider directly the greatest of the depths. Notice that the norm defined with respect to the empty order is, dually, a lower bound for the norms defined with respect to the other priority orders. Secondly, this choice allows us to give alternative formulations of both notions by induction on the structure of processes.

Definition 1 (Depth and norm)**.** The *depth* of a process is defined inductively on its structure by

- $\mathrm{depth}\,(a) = 1$;
- $\mathrm{depth}\,(p_1 \cdot p_2) = \mathrm{depth}\,(p_1) + \mathrm{depth}\,(p_2)$;
- $\mathrm{depth}\,(p_1 + p_2) = \max\{\mathrm{depth}\,(p_1)\,, \mathrm{depth}\,(p_2)\}$;
- $\mathrm{depth}\,(\Theta(p)) = \mathrm{depth}\,(p)$.

Similarly, the *norm* of process is defined inductively on its structure by

- $\mathrm{norm}\,(a) = 1$;
- $\mathrm{norm}\,(p_1 \cdot p_2) = \mathrm{norm}\,(p_1) + \mathrm{norm}\,(p_2)$;
- $\mathrm{norm}\,(p_1 + p_2) = \min\{\mathrm{norm}\,(p_1)\,, \mathrm{norm}\,(p_2)\}$;
- $\mathrm{norm}\,(\Theta(p)) = \mathrm{norm}\,(p)$.

Both notions can be extended to *process terms* by adding, respectively, the value of the depth and norm of a variable which are defined as $\mathrm{depth}\,(x) = 1$ and $\mathrm{norm}\,(x) = 1$.

We remark that although variables cannot perform any transition, as one can easily see from the inference rules in Table 1, their depth, and norm, are set to 1, since the minimal closed instance of a variable with respect to these measures is as a constant in $\mathcal{A}$.

For $p \in \mathbf{P}$, the set of *initial actions* of p with respect to $>$ is defined as

$$\mathrm{init}_>(p) = \{a \mid p \xrightarrow{a}_> p', p' \in \mathbf{P}\} \cup \{a \mid p \xrightarrow{a}_> \surd\}.$$

We extend this notion to sequences of transitions by letting $\mathrm{init}_>^k(p) = \bigcup_{p \to_>^k p'} \mathrm{init}_>(p')$ and $\mathrm{init}_>^\omega(p) = \bigcup_{k\in\mathbb{N}} \mathrm{init}_>^k(p)$ be, respectively, the set of actions that are enabled with respect to $>$ at depth k and at some depth. We say that action a is *maximal* with respect to $>$ if there is no $b \in \mathcal{A}$ such that $b > a$. We can restrict this notion to the set of actions that are enabled for a process. Given a process p, we say that an action $a \in \mathrm{init}_>^\omega(p)$ is *maximal in* p, or *locally maximal*, with respect to $>$ if there is no $b \in \mathrm{init}_>^\omega(p)$ such that $b > a$. If $\mathrm{init}_>^\omega(p) = \{a\}$ then a is locally maximal with respect to $>$.

2.2. Order-insensitive bisimulation

With the priority operator, the set of transitions that are enabled for each process depends on the considered priority order on $\mathcal{A}$. Therefore, any bisimulation relation over BPA_Θ processes will also depend on the priority order. In [3], along all such bisimulations, the authors introduced the notion of *order-insensitive bisimilarity*, $\underline{\leftrightarrow}_*$, formally defined as the intersection over all priority orders of the related bisimulation relations. Since $\underline{\leftrightarrow}_*$ disregards the particular order that is considered, it can be used to study general properties of processes and thus develop a general equational theory for BPA_Θ.

Definition 2 (Order-insensitive bisimulation, [3])**.** Let $>$ be any priority order. A binary symmetric relation $\mathcal{R} \subseteq \mathbf{P} \times \mathbf{P}$ is a *bisimulation with respect to* $>$ if whenever $p\,\mathcal{R}\,q$ then

- for all $p \xrightarrow{a}_> p'$ there is $q \xrightarrow{a}_> q'$ such that $p'\,\mathcal{R}\,q'$, and
- for all $p \xrightarrow{a}_> \surd$ also $q \xrightarrow{a}_> \surd$ holds.

$$(e_1)\ \frac{}{t \approx t} \qquad (e_2)\ \frac{t \approx u}{u \approx t} \qquad (e_3)\ \frac{t \approx u \quad u \approx v}{t \approx v} \qquad (e_4)\ \frac{t \approx u}{\sigma(t) \approx \sigma(u)}$$

$$(e_5)\ \frac{t_1 \approx u_1 \quad t_2 \approx u_2}{t_1 \cdot t_2 \approx u_1 \cdot u_2} \qquad (e_6)\ \frac{t_1 \approx u_1 \quad t_2 \approx u_2}{t_1 + t_2 \approx u_1 + u_2} \qquad (e_7)\ \frac{t \approx u}{\Theta(t) \approx \Theta(u)}$$

Table 2: Rules of equational logic over BPA_Θ.

We say that p, q are *bisimilar with respect to* $>$, denoted by $p \underline{\leftrightarrow}_> q$, if $p \mathcal{R} q$ holds for some bisimulation $\mathcal{R}$ with respect to $>$.

We say that p, q are *order-insensitive bisimilar*, denoted by $p \underline{\leftrightarrow}_* q$, if $p \underline{\leftrightarrow}_> q$ holds for all priority orders.

For a given priority order $>$, the bisimulation equivalence $\underline{\leftrightarrow}_>$ behaves like a classic bisimulation and therefore the following lemma, from [3], holds.

Lemma 1 ([3, Proposition 9])**.** *Consider processes p, q, assume $p \underline{\leftrightarrow}_> q$ for some priority order $>$ over $\mathcal{A}$, and let $k \in \mathbb{N}$. Then:*

1. *For every process p' such that $p \to_>^k p'$, there is a process q' such that $q \to_>^k q'$ and $p' \underline{\leftrightarrow}_> q'$.*
2. $\mathrm{init}_>^k(p) = \mathrm{init}_>^k(q)$ *so, in particular,* $\mathrm{init}_>^1(p) = \mathrm{init}_>^1(q)$.

It is not hard to prove that, since the inference rules in Table 1 respect the GSOS format [13], $\underline{\leftrightarrow}_>$ and $\underline{\leftrightarrow}_*$ are congruences over BPA_Θ processes. However, as discussed in [3], $\underline{\leftrightarrow}_*$ does not inherit the coinductive nature of bisimilarity, as we show in the following example.

***Example* 1.** Consider the processes $p = a \cdot b + a \cdot c + a \cdot (b + c)$ and $q = p + a \cdot \Theta(b + c)$. Notice that

- if $b > c$ then $a \cdot \Theta(b + c) \underline{\leftrightarrow}_> a \cdot b$,
- if $c > b$ then $a \cdot \Theta(b + c) \underline{\leftrightarrow}_> a \cdot c$, and
- if b, c are incomparable with respect to $>$ then $a \cdot \Theta(b + c) \underline{\leftrightarrow}_> a \cdot (b + c)$.

Therefore, we have that $p \underline{\leftrightarrow}_* q$. However, $q \xrightarrow{a}_> \Theta(b + c)$ for each order $>$, but there is no p' such that $p \xrightarrow{a}_> p'$ and $p' \underline{\leftrightarrow}_* \Theta(b + c)$. ◀

For sake of notation, henceforth, whenever $>$ is the empty order, we simply omit the subscript, i.e., $\to_\emptyset, \underline{\leftrightarrow}_\emptyset$ and $\mathrm{init}_\emptyset(\cdot)$ become, respectively, $\to, \underline{\leftrightarrow}$ and $\mathrm{init}(\cdot)$.

2.3. Equational logic

An *axiom system* $\mathcal{E}$ is a collection of *process equations* $t \approx u$ over the language BPA_Θ, such as those presented in Table 3. An equation $t \approx u$ is *derivable* from an axiom system $\mathcal{E}$, notation $\mathcal{E} \vdash t \approx u$, if there is an *equational proof* for it from $\mathcal{E}$, namely if it can be inferred from the axioms in $\mathcal{E}$ using the *rules* of *equational logic*, which are reflexivity, symmetry, transitivity, substitution and closure under BPA_Θ contexts, and are reported in Table 2.

Let $\mathcal{E}$ be a sound set of axioms. Rules (e_1)-(e_4) are common for all process languages and they ensure that $\mathcal{E}$ is closed with respect to reflexivity, symmetry, transitivity and substitution, respectively. Rules (e_5)-(e_7) are tailored for BPA_Θ and they ensure the closure of $\mathcal{E}$ under BPA_Θ contexts. They are therefore referred to as the *congruence rules.* Briefly, rule (e_5) is the rule for sequential composition and it states that whenever $\mathcal{E} \vdash t_1 \approx u_1$ and $\mathcal{E} \vdash t_2 \approx u_2$, then we can infer $\mathcal{E} \vdash t_1 \cdot u_1 \approx t_2 \cdot u_2$. Rule (e_6) deals with the nondeterministic choice operator in a similar way and rule (e_7) ensures that the priority operator preserves the equivalence of terms.

C1	$x + y \approx y + x$	S1	$(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$
C2	$(x + y) + z \approx x + (y + z)$	S2	$(x + y) \cdot z \approx (x \cdot z) + (y \cdot z)$
C3	$x + x \approx x$		
P1	$\Theta(\Theta(x) + y) \approx \Theta(x + y)$		
P2	$\Theta(x) + \Theta(y) \approx \Theta(x) + \Theta(y) + \Theta(x + y)$		
P3	$\Theta(x \cdot y) \approx \Theta(x) \cdot \Theta(y)$		
P4	$\Theta(x \cdot y + x \cdot z + w) \approx \Theta(x \cdot y + w) + \Theta(x \cdot z + w)$		
P5	$\Theta(a) \approx a$		

Table 3: Some axioms of BPA_Θ.

As elsewhere in the literature, we assume, without loss of generality, that for each axiom in $\mathcal{E}$ also the symmetric counterpart is in $\mathcal{E}$, so that the symmetry rule is not necessary in the proofs, and that substitution rules are always applied first in equational proofs, which means that the substitution rule $\dfrac{t \approx u}{\sigma(t) \approx \sigma(u)}$ may only be used for axioms $t \approx u$ in $\mathcal{E}$. If this is the case, then $\sigma(t) \approx \sigma(u)$ is called a *substitution instance* of the axiom.

The process equation $t \approx u$ is said to be *sound* with respect to $\underline{\leftrightarrow}_*$ if $\sigma(t) \underline{\leftrightarrow}_* \sigma(u)$ for all closed substitutions σ. For simplicity, if $t \approx u$ is sound, then we write $t \underline{\leftrightarrow}_* u$. An axiom system is *sound* modulo $\underline{\leftrightarrow}_*$ if and only if all of its equations are sound modulo $\underline{\leftrightarrow}_*$. Conversely, we say that $\mathcal{E}$ is *ground-complete* modulo $\underline{\leftrightarrow}_*$ if $p \underline{\leftrightarrow}_* q$ implies $\mathcal{E} \vdash p \approx q$ for all processes p, q. We say that $\underline{\leftrightarrow}_*$ is *finitely based*, if there is a *finite* axiom system $\mathcal{E}$ such that $\mathcal{E} \vdash t \approx u$ if and only if $t \underline{\leftrightarrow}_* u$. Finally, notice that the notion of depth can be extended to equations by letting $\mathrm{depth}\,(t \approx u) = \max\{\mathrm{depth}\,(t)\,, \mathrm{depth}\,(u)\}$.

3. Towards a negative result

As disclosed in the Introduction, our order of business for the remainder of this paper will be to prove the following theorem:

Theorem 1. *If the set of actions $\mathcal{A}$ contains at least two distinct actions, then the language BPA_Θ modulo order-insensitive bisimilarity is not finitely based.*

Due to the heavy amount of technical results that are needed to fulfill this purpose, we decided to dedicate this section to an informal description of our proof strategy. Hopefully, this will improve the readability of our paper and work as a guide for the reader in their journey through the technical development of our results.

3.1. The idea

Our method stems from [22–24], in which Moller discussed the axiomatiazability of the parallel composition operator and proved that (a fragment of) CCS modulo bisimilarity is not finitely based. The key idea is to identify a *special property* of BPA_Θ terms, say $\mathbb{P}(n)$ for $n \geq 0$, that, when n is *large enough*, is preserved by provability under finite axiom systems. Roughly, this means that if $\mathcal{E}$ is a finite set of axioms that are sound modulo order-insensitive bisimilarity, the equation $p \approx q$ is provable from $\mathcal{E}$, and n is greater than the depth of the equations in $\mathcal{E}$, then either both p and q satisfy $\mathbb{P}(n)$, or none of them does. Then we introduce a family of infinitely many equations $\{e_n \mid n \geq 0\}$ that are all sound modulo $\underline{\leftrightarrow}_*$, but are such that only one side of e_n satisfies $\mathbb{P}(n)$, for each $n \geq 0$. This implies that the family of equations cannot be derived from any finite axiom system that is sound modulo $\underline{\leftrightarrow}_*$ and, hence, at least infinitely many of those equations must be included in the axiomatisation, which is therefore *not* finitely based.

3.2. The choice of $\mathbb{P}(n)$

The property $\mathbb{P}(n)$ will involve the priority operator. We shall say, in a very informal way, that $\mathbb{P}(n)$ will be satisfied by a process p if it reaches, through a sequence of n *steps*, a process, say p', whose behaviour is *determined by* Θ. Intuitively, this means that p' behaves differently under different priority orders. For instance, p' could be of the form $\Theta(\Theta(\Theta(a) + b \cdot p''))$ for some $a \neq b$ and process p''. Then p' affords an a-transition and no b-transition if $a > b$, whereas p' affords a b-transition and no a-transition if $b > a$. It is important that $\mathcal{A}$ contains at least two actions, so that we can have different priority orders (possibly) triggering different behaviours of Θ-terms. Moreover, p' must have (a nesting of) Θ as head operator and a nondeterministic choice between (at least) two processes having distinct sets of initial actions must occur within the scope of such (nesting of) Θ.

Borrowing the terminology from [3], we will call Θ-*dependent* the process terms whose initial behaviour depends on the priority order. The choice of involving Θ-dependent terms in $\mathbb{P}(n)$ is strongly related to the fact that we are considering order-insensitive bisimilarity. In fact, as we need to take into account the behaviour of processes with respect to *all* priority orders, then no axiom can be used to eliminate the head occurrence of Θ from Θ-dependent terms. These terms and their properties will be presented in Section 6.

There is, however, another feature of order-insensitive bisimilarity that we will need to take into account to properly define the property $\mathbb{P}(n)$. As previously outlined, differently from classic notions of bisimulations, $\underline{\leftrightarrow}_*$ does not have, in general, a coinductive construction. Hence, to simplify the reasoning in the proofs, we need to define $\mathbb{P}(n)$ in such a way that only those processes on which $\underline{\leftrightarrow}_*$ can be defined coinductively could satisfy it. To this end we introduce, in Section 5, the notion of *uniform determinacy* as a sufficient condition to ensure the coinductive behavior of $\underline{\leftrightarrow}_*$.

The special property $\mathbb{P}(n)$ is then defined, in Section 6, as the property of *uniform* (n, Θ)-*dependency* of processes, which combines the ideas of determinacy and Θ-dependency of processes and, in addition, will require that all the processes in the sequence of n steps leading to the Θ-dependent term have norm 1. This is to guarantee that no axiom for sequential composition can be used to rewrite such a sequence.

3.3. The choice of n

The choice of n *large enough* will play a fundamental role in proving that whenever p satisfies $\mathbb{P}(n)$ then so does q, especially in the case in which $p \approx q$ is derived by an application of the substitution rule of equational logic (rule (e_4) in Table 2). In this case, we have $p = \sigma(t)$ and $q = \sigma(u)$ for some closed substitution σ and BPA_Θ terms t, u such that $t \approx u \in \mathcal{E}$. Then, if n is *large enough*, which translates into n being greater than the depth of the equations in $\mathcal{E}$ (and thus of the depth of all the terms occurring in such equations), we can prove that the fact that p satisfies $\mathbb{P}(n)$ is due to the behaviour of the closed instance of some variable x occurring in t. We can also prove that for $t \approx u$ to be sound modulo $\underline{\leftrightarrow}_*$, whenever a variable x occurs in t then it must also occur in u. Actually, we are going to prove the stronger result that if such an occurrence of x in t is within the scope a priority operator, then so is the occurrence of x in u. Hence, we can infer that $\sigma(x)$ will trigger in $\sigma(u)$ the same behaviour that it induced in $\sigma(t)$, and thus that also $q = \sigma(u)$ will satisfy $\mathbb{P}(n)$.

To obtain all the results mentioned in this subsection it will be fundamental to study the decomposition of the behaviour of closed instances of terms with respect to the behavior of the closed instances of variables occurring in them. Section 4 is devoted to such an analysis.

3.4. The family of equations

Consider the processes $\{P_n\}_{n \in \mathbb{N}}$, defined as follows

$$\begin{aligned} & P_n = A_n(a) + A_n(b) + A_n(a+b) && (n \geq 0) \\ \text{where} \quad & A_0(p) = p && \\ \text{and} \quad & A_{n+1}(p) = a \cdot A_n(p) + a && (n \geq 0) \;. \end{aligned}$$

Intuitively, the process P_n must at the top level decide whether it will end up in a, b, or $a + b$ after n steps. After making this choice, it can take up to n a-transitions, and at each step it can choose whether to terminate or to continue. The possibility of termination at each step is crucial, since it means that the process cannot be written just with sequential composition modulo bisimilarity.

As we will formally prove in Section 7, the following family of infinitely many sound equations shows that order-insensitive bisimilarity is not finitely based over BPA_Θ

$$e_n : \quad P_n + A_n(\Theta(a+b)) \approx P_n \qquad (n \geq 0) \ . \tag{1}$$

Informally, each equation e_n is sound, because, according to which priority order is considered, $\Theta(a+b)$ will be bisimilar to a, b or $a+b$, and thus the two sides of e_n are order-insensitive bisimilar. However, process $A_n(\Theta(a+b))$ can be proved to be *uniformly* (n,Θ)*-dependent*, whereas P_n is not. We will argue that this implies that not all the equations in the family $\{e_n\}_{n\in\mathbb{N}}$ can be derived from a finite set of valid axioms, thus proving Theorem 1.

4. Relation between open and closed operational behaviour

Our purpose in the remainder of this paper is to verify whether the axiomatisation for order-insensitive bisimilarity is finitely based over BPA_Θ. To address this question it is fundamental to establish a correspondence between the behaviour of open terms and the semantics of their closed instances, with a special focus on the role of variables. In this section, we provide the notions and theoretical results necessary to establish the desired behavioural correspondence.

4.1. From open to closed transitions...

Assume a term t, a closed substitution σ, a process p, an action a and a priority order $>$. We aim at investigating how to derive a transition of the form $\sigma(t) \xrightarrow{a}_{>} p$, as well as a predicate $\sigma(t) \xrightarrow{a}_{>} \surd\!\!\!\surd$, from the behaviour of t and of $\sigma(x)$ for each variable x occurring in t. In particular we are interested in relating the *initial* behaviour of $\sigma(t)$ with the behaviour of closed instances of variables occurring in it.

The simplest case is a direct application of the operational semantics in Table 1: if action a is maximal with respect to $>$, then $\sigma(t) \xrightarrow{a}_{>} p$ can be inferred directly from $t \xrightarrow{a}_{>} t'$, for some term t' with $\sigma(t') = p$. In fact, the maximality of a guarantees that the execution of the a-transition cannot be prevented by any occurrence of the priority operator. A similar reasoning holds for transition predicates.

Lemma 2. *Let t, t' be process terms, let a be an action with maximal priority with respect to $>$. Then for all substitutions σ it holds that:*

1. *If $t \xrightarrow{a}_{>} \surd\!\!\!\surd$ then $\sigma(t) \xrightarrow{a}_{>} \surd\!\!\!\surd$.*
2. *If $t \xrightarrow{a}_{>} t'$ then $\sigma(t) \xrightarrow{a}_{>} \sigma(t')$.*

Next we deal with variables. It may be the case, for instance, that the term t is of the form $t = x \cdot u$ for some term u. Clearly, the behaviour of $\sigma(t)$, and thus the derivation of $\sigma(t) \xrightarrow{a}_{>} p$, will depend on the behaviour of $\sigma(x)$. However, the set of initial actions of $\sigma(t)$ does not depend, in general, solely on those of $\sigma(x)$, but also on the structure of the process into which x is mapped, and on the occurrence of x in t. For instance, for $t = x \cdot u$ we can distinguish two main situations:

(I) Suppose $\sigma(x) = a$, so that $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$. This would give $\sigma(t) \xrightarrow{a}_{>} p$ for $p = \sigma(u)$, namely p is a closed instance of a subterm of t. Therefore, the transition for $\sigma(t)$ could be expressed in terms of a closed instance of an open transition for t, as $t \rightarrow_{>} u$. However, notice that the action that is performed cannot be obtained from the term t as it depends solely on the substitution applied to x. Hence, we will need a formal way to express that the label of the transition depends on x.

(II) Suppose $\sigma(x) = a \cdot b$, so that $\sigma(x) \xrightarrow{a}_{>} b$. Clearly, $\sigma(t)$ will have to mimic such behaviour, and thus $\sigma(t) \xrightarrow{a}_{>} p$ with $p = b \cdot \sigma(u)$. Notice that process p subsumes *what's left* of the behaviour of $\sigma(x)$. Then the transition for $\sigma(t)$ cannot be inferred from a closed substitution instance of an open transition of the form $t \xrightarrow{a}_{>} t'$, since the structure of t' cannot be known until the substitution $\sigma(x)$ has occurred. Hence, we will need a formal way to express that to reach a subterm of t we need to follow a sequence of transitions performed by x.

$$
(a_1)\ \frac{}{x \xrightarrow{x_s}_{>} x_d} \qquad (a_2)\ \frac{}{x \xrightarrow{x}_{>} \checkmark\checkmark}
$$

$$
(a_3)\ \frac{t \xrightarrow{x_s}_{>} c}{t \cdot u \xrightarrow{x_s}_{>} c \cdot u} \qquad (a_4)\ \frac{t \xrightarrow{x}_{>} t'}{t \cdot u \xrightarrow{x}_{>} t' \cdot u} \qquad (a_5)\ \frac{t \xrightarrow{x}_{>} \checkmark\checkmark}{t \cdot u \xrightarrow{x}_{>} u}
$$

$$
(a_6)\ \frac{t \xrightarrow{x_s}_{>} c}{t + u \xrightarrow{x_s}_{>} c} \qquad (a_7)\ \frac{t \xrightarrow{x}_{>} t'}{t + u \xrightarrow{x}_{>} t'} \qquad (a_8)\ \frac{t \xrightarrow{x}_{>} \checkmark\checkmark}{t + u \xrightarrow{x}_{>} \checkmark\checkmark}
$$

$$
(a_9)\ \frac{t \xrightarrow{x_s}_{>} c}{\Theta(t) \xrightarrow{x_s}_{>} \Theta(c)} \qquad (a_{10})\ \frac{t \xrightarrow{x}_{>} t'}{\Theta(t) \xrightarrow{x}_{>} \Theta(t')} \qquad (a_{11})\ \frac{t \xrightarrow{x}_{>} \checkmark\checkmark}{\Theta(t) \xrightarrow{x}_{>} \checkmark\checkmark}
$$

Table 4: Inference rules for the auxiliary transition relations. The symmetric versions of rules a_6–a_8 have been omitted.

For a formal development of the analysis in the above-mentioned cases, we exploit the method proposed in [5] and provide an auxiliary operational semantics tailored for expressing the behaviour of process terms resulting from that of closed substitution instances for their variables.

Firstly we introduce the notion of *configuration* over BPA_Θ terms, which stems from [5]. Configurations are terms defined over a set of variables $\mathcal{V}_\mathrm{d} = \{x_d \mid x \in \mathcal{V}\}$, disjoint from $\mathcal{V}$, and BPA_Θ terms. We use the variable x_d to express that the closed instance of x has started its execution, but has not terminated yet.

Definition 3 (BPA_Θ configuration)**.** The collection of BPA_Θ configurations is given by:

$$
c ::= t \mid x_d \mid c \cdot t \mid \Theta(c),
$$

where t is a BPA_Θ term and $x_d \in \mathcal{V}_\mathrm{d}$.

Notice that the grammar above guarantees that each configuration contains at most one occurrence of a variable in $\mathcal{V}_\mathrm{d}$, say x_d, and if such occurrence is in the scope of sequential composition, then x_d must occur as the first symbol in the composition.

Define the set of variable labels $\mathcal{V}_\mathrm{s} = \{x_s \mid x \in \mathcal{V}\}$, disjoint from $\mathcal{V}$, and assume any priority order $>$. We then introduce two auxiliary relations $\xrightarrow{x_s}_{>}$, $\xrightarrow{x}_{>}$, and the auxiliary predicate $\xrightarrow{x}_{>}\checkmark\checkmark$, whose operational semantics is given in Table 4. These allow us to express how the initial behaviour of a term can be derived from that of the variables occurring in it. Informally, the labels allow us to identify the variable that induces a particular transition. Transitions of the form $t \xrightarrow{x}_{>} t'$ and predicates $t \xrightarrow{x}_{>}\checkmark\checkmark$ allow us to deal with the case described in item (I) above. Conversely, transitions $t \xrightarrow{x_s}_{>} c$ are used for the case in item (II). The configuration c stores the *yet-to-terminate* behaviour of $\sigma(x)$. As an example, for the terms in item (II) we would have $c = x_d \cdot u$, and, since $\sigma(x) \xrightarrow{a}_{>} b$, we would let $\sigma[x_d \mapsto b](c) = b \cdot \sigma(u)$.

The following lemma formalizes the intuitions above. To avoid conflicts with any possible occurrence of the priority operator, we focus only on transitions labeled with actions that are (locally) maximal with respect to the chosen priority operator $>$. This type of transition will be sufficient for our purposes in the rest of the paper.

Lemma 3. *Let t be a process term, x a variable, σ a substitution and $a \in \mathcal{A}$ be maximal with respect to $>$. Then:*

1. *If $t \xrightarrow{x}_{>}\checkmark\checkmark$ and $\sigma(x) \xrightarrow{a}_{>}\checkmark\checkmark$, then $\sigma(t) \xrightarrow{a}_{>}\checkmark\checkmark$.*
2. *If $t \xrightarrow{x}_{>} t'$ and $\sigma(x) \xrightarrow{a}_{>}\checkmark\checkmark$, then $\sigma(t) \xrightarrow{a}_{>} \sigma(t')$.*
3. *If $t \xrightarrow{x_s}_{>} c$ and $\sigma(x) \xrightarrow{a}_{>} p$ for some process p, then $\sigma(t) \xrightarrow{a}_{>} \sigma[x_d \mapsto p](c)$.*

Proof. The proof proceeds by induction over the derivation of the considered auxiliary predicates and transitions and can be found in Appendix A.1. $\square$

We will sometimes need to extend the third case of Lemma 3 to sequences of transitions. To this end, we provide first an auxiliary technical lemma, that will simplify our reasoning.

Lemma 4. *Let $a \in \mathcal{A}$ be maximal with respect to $>$, and let σ be a closed substitution. Consider a configuration c, and processes p, p' such that $p \xrightarrow{a}_{>} p'$. If c contains an occurrence of x_d, then $\sigma[x_d \mapsto p](c) \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](c)$.*

Proof. The proof proceeds by structural induction over the configuration c and can be found in Appendix A.2. □

We can now show that the decomposition of the semantics can be extended to sequences of transitions, and we can thus apply inductive arguments to them.

Lemma 5. *Let σ be a closed substitution. If $t \xrightarrow{x_s}_{>} c$ and $\sigma(x) \rightarrow^n_{>} p$ is such that all actions taken along the transitions from $\sigma(x)$ to p are maximal with respect to $>$, then $\sigma(t) \rightarrow^n_{>} \sigma[x_d \mapsto p](c)$.*

Proof. The proof proceeds by a simultaneous induction over the derivation of the auxiliary transition $t \xrightarrow{x_s}_{>} c$ and over $n \in \mathbb{N}$, and can be found in Appendix A.3. □

4.2. ... and back again

So far we have provided a way to derive the initial behaviour of a term from the open transitions available for it, especially when determined by variables. Our aim is now to obtain a converse result: knowing that $\sigma(t) \xrightarrow{a}_{>} p$, we want to charcterise its possible sources in the behaviour of t and of the closed instances of the variables occurring in t.

Firstly, we remark that in Section 4.1 we have considered *open* process terms and thus no occurrence of a priority operator, due to substitutions of variables possibly occurring in them, could have been foreseen. Therefore, to avoid conflicts, we have limited our attention to actions that were (locally) maximal with respect to the considered priority order. However, we now start from the closed process term $\sigma(t)$ and therefore we can properly relate the behaviour of the closed instances of variables to their potential occurrence in the scope of a priority operator. To this end, we introduce the relation of *initial enabledness* between a variable x and a term t with respect to a natural number $l \in \mathbb{N}$, notation $x \lhd_l t$. Informally, $x \lhd_l t$ holds if x occurs in the scope of l-nested applications of the priority operator in t and the initial behaviour of $\sigma(t)$ is possibly determined by $\sigma(x)$, for all substitutions σ. Initial enabledness extends relation $\lhd_l$ from [3], that was defined on BCCSP_Θ terms, to BPA_Θ terms.

Definition 4 (Initial enabledness, $\lhd_l$)**.** The relations $\lhd_l$, for $l \in \mathbb{N}$, between variables and terms are defined as the least relations satisfying the following constraints:

1. $x \lhd_0 x$;
2. if $x \lhd_l t$ then $x \lhd_l t + u$ and $x \lhd_l u + t$;
3. if $x \lhd_l t$ then $x \lhd_l t \cdot t'$;
4. if $x \lhd_l t$ then $x \lhd_{l+1} \Theta(t)$.

If $x \lhd_l t$, for some $l \in \mathbb{N}$, we say that x is *initially enabled* in t. We say that x is *initially disabled* in t, otherwise.

***Example* 2.** Consider the terms $t_1 = x \cdot \Theta(u_1)$, for some term u_1 such that $x \notin \mathrm{var}(u_1)$, and $t_2 = \Theta(\Theta(\Theta(t_1 + u_2) \cdot y)) \cdot u_3$, for some variable $y \neq x$ and terms u_2, u_3, such that $x \notin \mathrm{var}(u_2), \mathrm{var}(u_3)$. Then we have that $x \lhd_0 t_1$, $x \lhd_0 t_1 + u_2$ and $x \lhd_3 t_2$, so that x is initially enabled in t_1, $t_1 + u_2$ and t_2.

Conversely, variable y is initially disabled in t_2 as it occurs as second argument of a sequential composition operator. Notice that this implies that no action performed by any closed substitution instance of y can trigger a transition of the corresponding closed instance of t_2. ◀

As stated by the following lemma, there is a close relation between x being initially enabled in t and the auxiliary transition $t \xrightarrow{x_s}_{>} c$. We write $t = t_1 \odot t_2$ to mean that either $t = t_1$ or $t = t_1 \cdot t_2$, i.e., t_1 may possibly be sequentially followed by t_2. We extend this notation to nested occurrences of possible sequential compositions $\odot$ by $t \bigodot_{i=1}^{n} t_i = (\dots(t \odot t_1) \odot \dots) \odot t_n$. Then, for a process term t and $l \in \mathbb{N}$ we define the set of terms $\Theta^l_{\odot}(t)$ inductively as follows:

$$\Theta^0_{\odot}(t) = \left\{ u \mid u = t \bigodot_{i=1}^{n} t_i \text{ for some } n \in \mathbb{N} \text{ and terms } t_1, \dots, t_n \right\}$$

$$\Theta^{l+1}_{\odot}(t) = \left\{ u \mid u = \Theta(u' \odot t') \bigodot_{i=1}^{n} t_i \text{ for some } u' \in \Theta^l_{\odot}(t), n \in \mathbb{N} \text{ and terms } t', t_1, \dots, t_n \right\} .$$

In what follows, we write $t \rightarrow_{>} \Theta^l_{\odot}(t')$ to denote that $t \rightarrow_{>} u$ for some $u \in \Theta^l_{\odot}(t')$. Substitutions and transitions are lifted to $\Theta^l_{\odot}(t)$ in a similar fashion.

Lemma 6. *Let x be a variable, t a term and $l \in \mathbb{N}$. Then, $x \lhd_l t$ if and only if $t \xrightarrow{x_s}_{>} \Theta^l_{\odot}(x_d)$.*

Proof. The proof can be found in Appendix A.4. □

The notation $\Theta^l_{\odot}(x_d)$ abstracts away from a tail of nested (possible) sequential compositions. This choice is merely for simplification purposes and does not impact the technical development of our results. In fact, the behaviour of the terms in the tail and their closed instances will never play a role in the results, as only the contribution of closed instances of x_d to the behaviour of terms in $\Theta^l_{\odot}(x_d)$ will be of interest. We remark also that $\Theta^0_{\odot}(x_d)$ denotes a configuration containing an occurrence of x_d which is not in the scope of a priority operator.

***Example* 3.** Consider the terms t_1, t_2 in Example 2 and assume a priority order $>$. Since $x \xrightarrow{x_s}_{>} x_d$, by rule (a_3) in Table 4 we get $t_1 \xrightarrow{x_s}_{>} x_d \cdot \Theta(u_1)$ which, by rule (a_6) in Table 4, gives $t_1 + u_2 \xrightarrow{x_s}_{>} x_d \cdot \Theta(u_1)$. Hence, by three applications of rule (a_9) and as many of rule (a_3), we infer that $t_2 \xrightarrow{x_s}_{>} \Theta(\Theta(\Theta(x_d \cdot \Theta(u_1)) \cdot y)) \cdot u_3$. Notice that the right-hand side of the transition from t_2 is of the form $\Theta^3_{\odot}(x_d)$ and that the trailing $\Theta(u_1), y, u_3$ played no role in the derivation of such a transition. ◀

We are now ready to derive the behaviour of the term t and that of the closed instances of the variables occurring in t, from the transitions enabled for $\sigma(t)$.

Proposition 1. *Let t be a process term, σ a closed substitution, a an action and p a process. Then:*

1. *If $\sigma(t) \xrightarrow{a}_{>} \surd$ then*
   - *(a) either $t \xrightarrow{a}_{>} \surd$;*
   - *(b) or there is a variable x such that $t \xrightarrow{x}_{>} \surd$ and $\sigma(x) \xrightarrow{a}_{>} \surd$.*
2. *If $\sigma(t) \xrightarrow{a}_{>} p$ then one of the following applies:*
   - *(a) there is a process term t' such that $t \xrightarrow{a}_{>} t'$ and $\sigma(t') = p$;*
   - *(b) there are a process term t' and a variable x such that $t \xrightarrow{x}_{>} t'$, $\sigma(x) \xrightarrow{a}_{>} \surd$ and $\sigma(t') = p$;*
   - *(c) there are a variable x, a natural number $l \in \mathbb{N}$, and a process q such that $t \xrightarrow{x_s}_{>} \Theta^l_{\odot}(x_d)$, $\sigma(x) \xrightarrow{a}_{>} q$ and $p \in \Theta^l_{\odot}(q)$.*

Proof. 1. We proceed by induction over the derivation of $\sigma(t) \xrightarrow{a}_{>} \surd$.

- Base case: the last rule applied in the derivation of $\sigma(t) \xrightarrow{a}_{>} \surd$ is (r_1) in Table 1. This means that either $t = a$, or $t = x$ with $\sigma(x) = a$. In the former case it follows that $t \xrightarrow{a}_{>} \surd$ by rule (r_1) in Table 1 and in the latter it follows that $t \xrightarrow{x}_{>} \surd$ by rule (a_2) in Table 4 and $\sigma(x) \xrightarrow{a}_{>} \surd$.

- Inductive step $t = t_1 + t_2$ and $\sigma(t) \xrightarrow{a}_> \surd\!\!\surd$ is derived either by rule (r_4) in Table 1, and thus by $\sigma(t_1) \xrightarrow{a}_> \surd\!\!\surd$, or by rule (r_5) in Table 1, and thus by $\sigma(t_2) \xrightarrow{a}_> \surd\!\!\surd$. Assume, without loss of generality, that rule (r_4) was applied. By induction over $\sigma(t_1) \xrightarrow{a}_> \surd\!\!\surd$ we can distinguish two cases:
  - $t_1 \xrightarrow{a}_> \surd\!\!\surd$. Then by rule (r_4) in Table 1 we derive that $t \xrightarrow{a}_> \surd\!\!\surd$.
  - There is a variable x such that $t_1 \xrightarrow{x}_> \surd\!\!\surd$ and $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$. Hence, by applying rule (a_8) in Table 4 we derive that, for the same variable x, $t \xrightarrow{x}_> \surd\!\!\surd$.
- Inductive step: $t = \Theta(u)$ and $\sigma(t) \xrightarrow{a}_> \surd\!\!\surd$ is derived by rule (r_8) in Table 1. This implies that $\sigma(u) \xrightarrow{a}_> \surd\!\!\surd$ and $\sigma(u) \overset{b}{\not\to}_>$ for all $b > a$. By induction over $\sigma(u) \xrightarrow{a}_> \surd\!\!\surd$ we can distinguish two cases:
  - $u \xrightarrow{a}_> \surd\!\!\surd$. Since moreover from $\sigma(u) \overset{b}{\not\to}_>$ for all $b > a$ we can infer that $u \overset{b}{\not\to}_>$ for all such b, the premises of rule (r_8) in Table 1 are satisfied and we can derive that $t \xrightarrow{a}_> \surd\!\!\surd$.
  - There is a variable x such that $u \xrightarrow{x}_> \surd\!\!\surd$ and $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$. By applying rule (a_{11}) in Table 4 we derive that, for the same variable, $t \xrightarrow{x}_> \surd\!\!\surd$.

2. We proceed by induction over the derivation of $\sigma(t) \xrightarrow{a}_> p$. Hence, we assume that the property in Proposition 1.2 has been proven for all proper subderivations of the derivation of $\sigma(t) \xrightarrow{a}_> p$. We proceed by a case analysis over the structure of t to prove that the desired property holds for $\sigma(t) \xrightarrow{a}_> p$ as well. Notice that the case $t = a$ is vacuous, since there is no closed term p such that $a \xrightarrow{a}_> p$.
   - Case: $t = x$. Then case (2c) is satisfied directly by rule (a_1) in Table 4.
   - Case: $t = t_1 \cdot t_2$. We can distinguish two cases:
     - $\sigma(t) \xrightarrow{a}_> p$ is derived by rule (r_2) in Table 1, namely by $\sigma(t_1) \xrightarrow{a}_> \surd\!\!\surd$ and $p = \sigma(t_2)$. From $\sigma(t_1) \xrightarrow{a}_> \surd\!\!\surd$ and Proposition 1.1 we get that either $t_1 \xrightarrow{a}_> \surd\!\!\surd$ or there is a variable x such that $t_1 \xrightarrow{x}_> \surd\!\!\surd$ and $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$. In the former case we can apply rule (r_2) in Table 1 and obtain $t \xrightarrow{a}_> t_2$ with $\sigma(t_2) = p$, thus case (2a) is satisfied. In the latter case we can apply rule (a_5) in Table 4 and obtain $t \xrightarrow{x}_> t_2$ which together with $\sigma(t_2) = p$ and $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$ satisfies case (2b).
     - $\sigma(t) \xrightarrow{a}_> p$ is derived by rule (r_3) in Table 1, namely by $\sigma(t_1) \xrightarrow{a}_> p_1$ with $p_1 = q \cdot \sigma(t_2)$. By induction over $\sigma(t_1) \xrightarrow{a}_> p_1$ we can distinguish three cases:
       * Case (2a) applies so that there is a process term t_1' such that $t_1 \xrightarrow{a}_> t_1'$ and $\sigma(t_1') = p_1$. Then, by rule (r_3) in Table 1 we infer that $t \xrightarrow{a}_> t_1' \cdot t_2$ with $\sigma(t_1') \cdot \sigma(t_2) = p$, and thus case (2a) is also satisfied by t.
       * Case (2b) applies so that there are a process term t_1' and a variable x such that $t_1 \xrightarrow{x}_> t_1'$, $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$ and $\sigma(t_1') = p_1$. Then, by rule (a_4) in Table 4 we infer that $t \xrightarrow{x}_> t_1' \cdot t_2$ with $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$ and $\sigma(t_1') \cdot \sigma(t_2) = p$, and thus case (2b) is also satisfied by t.
       * Case (2c) applies so that there are a variable x, a natural $l \in \mathbb{N}$ and a process s such that $t_1 \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$, $\sigma(x) \xrightarrow{a}_> q$ and $p_1 \in \Theta^l_\odot(q)$. Notice that, since in the construction of $\Theta^l_\odot(x_d)$ we allow the nesting of trailing sequential components to be of arbitrary depth, we can infer that for all $u \in \Theta^l_\odot(x_d)$ the term $u \cdot t_2$ is also in $\Theta^l_\odot(x_d)$. Then, by rule (a_3) in Table 4 we infer that $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$. Hence case (2c) is also satisfied by t with respect to $\Theta^l_\odot(x_d)$, the variable x, the natural $l \in \mathbb{N}$ and the process q for which $p \in \Theta^l_\odot(q)$.
   - Case: $t = t_1 + t_2$ and $\sigma(t) \xrightarrow{a}_> p$ is derived either from $\sigma(t_1) \xrightarrow{a}_> p$ or$\sigma(t_2) \xrightarrow{a}_> p$, namely by applying either rule (r_6) or rule (r_7) in Table 1. Since induction applies to such a move taken by $\sigma(t_i)$ and in all the rules for nondeterministic choice in Tables 1 and 4 the moves of t_i are mimicked exactly by t, we can infer that each of the three cases of Proposition 1.2 holds for t whenever it holds for t_i.

- Case: $t = \Theta(u)$ and $\sigma(t) \xrightarrow{a}_{>} p$ is derived by applying rule (r_9) in Table 1. This implies that $\sigma(u) \xrightarrow{a}_{>} p_1$, with $\Theta(p_1) = p$, and $\sigma(u) \overset{b}{\not\to}_{>}$ for all $b > a$. By induction over $\sigma(u) \xrightarrow{a}_{>} p_1$ we can distinguish three cases:
  - Case (2a) applies so that there is a process term u' such that $u \xrightarrow{a}_{>} u'$ and $\sigma(u') = p_1$. Moreover, we remark that from $\sigma(u) \overset{b}{\not\to}_{>}$ for all $b > a$, it follows that $u \overset{b}{\not\to}_{>}$ for all $b > a$. Then, by rule (r_9) in Table 1 we infer that $t \xrightarrow{a}_{>} \Theta(u')$ with $\sigma(\Theta(u')) = p$, and thus case (2a) is also satisfied by t.
  - Case (2b) applies so that there are a process term u' and a variable x such that $u \xrightarrow{x}_{>} u'$, $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\surd$ and $\sigma(u') = p_1$. Then, by rule (a_{10}) in Table 4 we infer that $t \xrightarrow{x}_{>} \Theta(u')$ with $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\surd$ and $\sigma(\Theta(u')) = p$, and thus case (2b) is also satisfied by t.
  - Case (2c) applies so that there are a variable x, a natural $l \in \mathbb{N}$ and a process q such that $u \xrightarrow{x_s}_{>} \Theta^l_{\odot}(x_d)$, $\sigma(x) \xrightarrow{a}_{>} q$ and $p_1 \in \Theta^l_{\odot}(q)$. Now we notice that for each $u \in \Theta^l_{\odot}(x_d)$ it holds that $\Theta(u) \in \Theta^{l+1}_{\odot}(x_d)$. Then, by rule (a_9) in Table 4 we infer that $t \xrightarrow{x_s}_{>} \Theta^{l+1}_{\odot}(x_d)$. Hence case (2c) is also satisfied by t with respect to the variable x, the natural $l+1$ and the process q for which $p \in \Theta^{l+1}_{\odot}(q)$.

$\square$

Assume a process term t and suppose that depth$(t) = k$ for some $k \in \mathbb{N}$. We recall that the notion of depth as we have defined it in Definition 1 is with respect to the empty priority order. Clearly, given any closed substitution σ we will have that depth$(\sigma(t)) = n$ for some $n \geq k$. In particular, whenever n is *strictly* greater than k we can infer that at least one variable occurring in t has been mapped into a process defined using the sequential composition operator. Hence, we need to extend Proposition 1 to sequences of transitions of arbitrary length.

To this end, we introduce the following notation: let $w \in (\mathcal{A} \cup \mathcal{V})^*$ be a string $w = \alpha_1 \dots \alpha_h$ in which each α_i can be either an action or a variable. Then, given a substitution σ, we write $t \xrightarrow{s_1 \dots s_h}_{>,w} t'$ if there are process terms $t_0, \dots, t_h$ such that $t = t_0$, $t' = t_h$, and, for all $i \in \{1, \dots, h\}$,

- $s_i \in \mathcal{A}^*$;
- if $\alpha_i \in \mathcal{V}$, then $\sigma(\alpha_i) \xrightarrow{s_i}_{>} \surd\!\!\surd$ and $t_{i-1} \xrightarrow{s_i}_{>} t_i$;
- if $\alpha_i \in \mathcal{A}$, then $s_i = \alpha_i$ and $t_{i-1} \xrightarrow{\alpha_i}_{>} t_i$.

Finally, we write $|s_1 \dots s_h|$ for the *length* of $s_1 \dots s_h$.

***Example* 4.** Consider the term $t = a \cdot b \cdot x \cdot u$, for some term u, and the strings $w_1 = ab$ and $w_2 = abx$. Clearly, as string w_1 only considers the execution of a particular sequence of actions, we can write $t \xrightarrow{ab}_{>,w_1} x \cdot u$ since $t \xrightarrow{a}_{>} b \cdot x \cdot u \xrightarrow{b}_{>} x \cdot u$. Conversely, string w_2 requires *concatenating* the first two steps of t with the behavior of the variable x. Assume, for instance, a closed substitution σ with $\sigma(x) = a \cdot a \cdot b$, namely $\sigma(x) \xrightarrow{aab}_{>} \surd\!\!\surd$. Then, for the chosen substitution, we can *unfold* the behaviour of x in that of t, and write $t \xrightarrow{abaab}_{>,w_2} u$. ◀

We also notice that by Lemma 4, if $p \xrightarrow{a}_{>} p'$ for some action a having (locally) maximal priority with respect to $>$, then $\sigma[x_d \mapsto p](\Theta^l_{\odot}(x_d)) \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](\Theta^l_{\odot}(x_d))$. In this case, we abuse notation slightly and write directly $\Theta^l_{\odot}(p) \xrightarrow{a}_{>} \Theta^l_{\odot}(p')$.

Proposition 2. *Let t be a process term, σ a closed substitution, $n \in \mathbb{N}$ and p a process. If $\sigma(t) \to^n_{>} p$ then:*

1. *there exist a process term t', a string $w \in (\mathcal{A} \cup \mathcal{V})^*$ and $s_1 \dots s_h \in \mathcal{A}^*$ such that $t \xrightarrow{s_1 \dots s_h}_{>,w} t'$, $\sigma(t') = p$, and $|s_1 \dots s_h| = n$;*

2. or $t \xrightarrow{s_1 \ldots s_h}_{>,w} t'$ for some $w \in (\mathcal{A} \cup \mathcal{V})^*$ and $s_1 \ldots s_h$ such that $|s_1 \ldots s_h| = k < n$, and there are a variable x, a natural number $l \in \mathbb{N}$ and a process q, such that $t' \xrightarrow{x_s}_{>} \Theta^l_{\odot}(x_d)$, $\sigma(x) \rightarrow^{n-k}_{>} q$ and $p \in \Theta^l_{\odot}(q)$.

Proof. We proceed by induction over n.

- Base case $n = 1$. This directly follows by Proposition 1.2.
- Inductive step $n > 1$. $\sigma(t) \rightarrow^n_> p$ is equivalent to writing $\sigma(t) \rightarrow_> p_1 \rightarrow^{n-1}_> p$, for some process p_1. We can assume without loss of generality that $\sigma(t) \xrightarrow{a}_> p_1$. According to Proposition 1.2, from $\sigma(t) \xrightarrow{a}_> p_1$ we can distinguish three cases:
  1. there is a process term t_1 such that $t \xrightarrow{a}_> t_1$ and $\sigma(t_1) = p_1$. Then by induction over $p_1 \rightarrow^{n-1}_> p$ we can distinguish two subcases:
     - there is $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$ with $t_1 \xrightarrow{s_1 \ldots s_h}_{>,w_1} t'$ such that $|s_1 \ldots s_h| = n-1$ and $\sigma(t') = p$. Then, the proof can be concluded by noticing that for the sequence $w = aw_1$ we get $t \xrightarrow{as_1 \ldots s_h}_{>,w} t'$ with $|as_1 \ldots s_h| = n$ and $\sigma(t') = p$.
     - there are $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$, a variable y, a natural $l \in \mathbb{N}$ and a process q, such that $t_1 \xrightarrow{s_1 \ldots s_h}_{>,w_1} t'$ with $|s_1 \ldots s_h| = k < n-1$, $t' \xrightarrow{y_s}_> \Theta^l_{\odot}(y_d)$, $\sigma(y) \rightarrow^{n-1-k}_> q$ and $p \in \Theta^l_{\odot}(q)$. Then, the proof can be concluded by noticing that for the sequence $w = aw_1$ we get $t \xrightarrow{as_1 \ldots s_h}_{>,w} t'$ with $|as_1 \ldots s_h| = k+1 < n$ and y, l, q behave as before.
  2. there are a process term t_1 and a variable x such that $t \xrightarrow{x}_> t_1$, $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$ and $\sigma(t_1) = p_1$. Then by induction over $p_1 \rightarrow^{n-1}_> p$ we can distinguish two subcases:
     - there is $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$ with $t_1 \xrightarrow{s_1 \ldots s_h}_> t'$ such that $|s_1 \ldots s_h| = n-1$ and $\sigma(t') = p$. Then, the proof can be concluded by noticing that for the sequence $w = xw_1$ we get $t \xrightarrow{as_1 \ldots s_h}_{>,w} t'$ with $|as_1 \ldots s_h| = n$, as $|a| = 1$, and $\sigma(t') = p$.
     - there are $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$, a variable y, a natural $l \in \mathbb{N}$ and a process q, such that $t_1 \xrightarrow{s_1 \ldots s_h}_{>,w_1} t'$ with $|s_1 \ldots s_h| = k < n-1$, $t' \xrightarrow{y_s}_> \Theta^l_{\odot}(y_d)$, $\sigma(y) \rightarrow^{n-1-k}_> q$ and $p \in \Theta^l_{\odot}(q)$. Then, the proof can be concluded by noticing that, since $\sigma(x) \xrightarrow{a}_> \surd\!\!\surd$ gives $|a| = 1$, for the sequence $w = xw_1$ we get $t \xrightarrow{as_1 \ldots s_h}_{>,w} t'$ with $|as_1 \ldots s_h| = k+1 < n$ and c, x, q behave as before.
  3. there are a variable x, a natural $l \in \mathbb{N}$ and a process p' such that $t \xrightarrow{x_s}_> \Theta^l_{\odot}(x_d)$, $\sigma(x) \xrightarrow{a}_> p'$ and $p_1 \in \Theta^l_{\odot}(p')$. Recall that, per assumption, $p_1 \rightarrow^{n-1}_> p$. Since, $p_1 \in \Theta^l_{\odot}(p')$, we have that either all, or part of, the transitions in the sequence $p_1 \rightarrow^{n-1}_> p$ are executed within the scope of a priority operator (unless $l = 0$, but then this case would be an instance of Proposition 2.1). Therefore, we are guaranteed that the actions labelling the transitions that are performed in the scope of Θ are all locally maximal with respect to $>$. Therefore, Lemma 5 allows us to distinguish two cases:
     - $\sigma(x) \rightarrow^h_> q$ for some $h \geq n$. In this case the proposition follows by taking the empty string for w and the process q' such that $\sigma(x) \rightarrow^n_> q'$ and $p \in \Theta^l_{\odot}(q')$.
     - $\sigma(x) \rightarrow^k_> q \rightarrow_> \surd\!\!\surd$ for some $k < n$. Notice that this implies that there is some string s_x with $|s_x| = k$ of actions that have been performed by $\sigma(x)$. Due to the structure of $\Theta^l_{\odot}(x_d)$ we can infer that there are a natural $m \in \mathbb{N}$ and a process term

       $$t_1 = \underbrace{\Theta(\cdots\Theta}_{m \text{ times}}(t'' \odot u_{m+1}) \odot u_m) \ldots) \odot u_1$$

       such that $\sigma(t) \rightarrow^k_> \sigma(t_1) = p_1$. Since then $p_1 \rightarrow^{n-k}_> p$, by induction we can distinguish two subcases:

* there is $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$ with $t_1 \xrightarrow{s_1 \dots s_h}_{>, w_1} t'$ such that $|s_1 \dots s_h| = n - k$ and $\sigma(t') = p$. Then, the proof can be concluded by noticing that for the sequence $w = xw_1$ we get $t \xrightarrow{s_x s_1 \dots s_h}_{>, w} t'$ with $|s_x s_1 \dots s_h| = n$, as $|s_x| = k$, and $\sigma(t') = p$.
* there are $w_1 \in (\mathcal{A} \cup \mathcal{V})^*$, a variable y, a process q', and $m' \in \mathbb{N}$, such that $t_1 \xrightarrow{s_1 \dots s_h}_{>, w_1} t'$ with $|s_1 \dots s_h| = j < n - k$, $t' \xrightarrow{y_s}_{>} \Theta_{\odot}^{m'}(y_d)$, $\sigma(y) \rightarrow_{>}^{n-k-j} q'$ and $p \in \Theta_{\odot}^{m'}(q')$. Then, the proof can be concluded by noticing that, as $|s_x| = k$, for the sequence $w = xw_1$ we get $t \xrightarrow{s_x s_1 \dots s_h}_{>, w} t'$ with $|s_x s_1 \dots s_h| = k + j < n$ and y, m', q' as above.

□

The following result allows us to establish whether the behaviour of two bisimilar process terms is determined by the same variable. Moreover, it guarantees that such a variable is initially enabled in one term if and only if it is initially enabled in the other one.

Theorem 2. *Assume that $\mathcal{A}$ contains at least two actions, a and b. Let x be a variable. Consider two process terms t and u such that* $\mathrm{init}^{\omega}(t) \subseteq \{a\}$ *and $t \mathrel{\underline{\leftrightarrow}_*} u$. Whenever there is t' such that $t \rightarrow^k t'$, for some $k \in \mathbb{N}$, and $x \lhd_l t'$, for some $l \in \mathbb{N}$, then there is u' such that $u \rightarrow^k u'$ and $x \lhd_m u'$ for some $m \in \mathbb{N}$. Moreover, $l = 0$ if and only if $m = 0$.*

Proof. Let $n \in \mathbb{N}$ be larger than the depths of t and u, and assume the priority order $> = \{(b, a)\}$ over $\mathcal{A}$. We define the family of closed substitutions $\{\sigma_i\}_{i \in \mathbb{N}}$ inductively as follows:

$$\sigma_0(y) = \begin{cases} a + b & \text{if } y = x \\ a & \text{otherwise.} \end{cases}$$

$$\sigma_i(y) = \begin{cases} a \cdot (\sigma_{i-1}(y) + a) & \text{if } y = x \\ a & \text{otherwise.} \end{cases}$$

Let $\sigma = \sigma_n$. Suppose that $t \rightarrow^k t'$, for some $k \in \mathbb{N}$. As $\mathrm{init}^{\omega}(t) \subseteq \{a\}$ we can infer that there are process terms $t_0, \dots, t_k$ such that $t = t_0 \xrightarrow{a} \dots \xrightarrow{a} t_k = t'$ (if $\mathrm{init}(t) = \emptyset$ then $k = 0$ and $t = t'$). Moreover, as in all such terms t_i there is no occurrence of b, a is maximal with respect to $>$ on them, and thus by Lemma 2 and an easy induction over k, we obtain that $\sigma(t_0) \xrightarrow{a}{}^k \sigma(t_k) = \sigma(t')$ ($\sigma(t) = \sigma(t')$ if $\mathrm{init}(t) = \emptyset$). Suppose now that $x \lhd_l t'$, for some $l \in \mathbb{N}$. By Lemma 6, $x \lhd_l t'$ implies that $t' \xrightarrow{x_s}_{>} \Theta_{\odot}^{l}(x_d)$. By the choice of σ we have that $\sigma(x) \xrightarrow{a}{}_{>}^{n} a + b$. Therefore, by Lemma 5 we obtain that $\sigma(t') \xrightarrow{a}{}_{>}^{n} p$ for some $p \in \Theta_{\odot}^{l}(a + b)$. By combining the two sequences of transitions, we get $\sigma(t) \xrightarrow{a}{}_{>}^{k+n} p$. By the hypothesis we have $t \mathrel{\underline{\leftrightarrow}_*} u$, which in particular implies $t \mathrel{\underline{\leftrightarrow}_>} u$ and thus $\sigma(t) \mathrel{\underline{\leftrightarrow}_>} \sigma(u)$. As $\underline{\leftrightarrow}_>$ is a bisimulation, we can infer that $\sigma(u) \xrightarrow{a}{}_{>}^{k+n} p'$ for some process p' with $p \mathrel{\underline{\leftrightarrow}_>} p'$. As n is larger than the depth of u, by Proposition 2 there exists a process term u', a string w with strings $s_1, \dots, s_h \in \{a\}^*$, a variable y, a natural number m and a process q such that $u \xrightarrow{s_1 \dots s_h}_{>, w} u'$, $|s_1 \dots s_h| = j < n$, $u' \xrightarrow{x_s}_{>} \Theta_{\odot}^{m}(y_d)$, $\sigma(y) \rightarrow_{>}^{k+n-j} q$ and $p' \in \Theta_{\odot}^{m}(q)$. Therefore: (i) by $k + n - j > 0$; (ii) by the choice of $>$ (which gives that the only possible transition enabled for $\Theta_{\odot}^{l}(a + b)$ is a b-labeled move); (iii) by the choice of σ; (iv) by $p \mathrel{\underline{\leftrightarrow}_>} p'$ with $p \in \Theta_{\odot}^{l}(a + b), p' \in \Theta_{\odot}^{m}(q)$; we can conclude that $y = x$, $j = k$ and $q = a + b$. Moreover, from item (iv) and the choice of $>$, we obtain that $l = 0$ iff $m = 0$. □

5. Making order-insensitive bisimilarity coinductive: uniform determinacy

As outlined in Section 2, $\underline{\leftrightarrow}_*$ cannot be defined coinductively, contrary to other bisimulation relations. However, in this section we identify a class of processes for which the coinductive reasoning on $\underline{\leftrightarrow}_*$ can be at least partially recovered, and which will be useful later on.

Definition 5 (Uniform determinacy)**.** Let p be a process. We say that p is *uniformly determinate* if $|\mathrm{init}(p)| = 1$, and for all processes p_1 and p_2 such that $p \rightarrow p_1$ and $p \rightarrow p_2$, we have $\mathrm{norm}\,(p_1) = \mathrm{norm}\,(p_2) = 1$ and $p_1 \mathrel{\underline{\leftrightarrow}_*} p_2$. Then, for each $k \in \mathbb{N}$, we say that p is *uniformly k-determinate* if

- $|\text{init}(p)| = 1$,
- whenever $p \to^h q$ for some $h \leq k$ then $|\text{init}(q)| = 1$, and
- whenever $p \to^k p'$ then p' is uniformly determinate.

We remark that uniform determinacy and uniform k-determinacy are defined in terms of the empty priority order.

Summarizing, a process is uniformly k-determinate if whenever it takes k steps, it ends up in a process that only has one available action, and in which all immediate successors have norm 1 and are order-insensitive bisimilar.

***Example* 5.** Consider processes

$$p_1 = a \cdot b + a \qquad p_2 = a \cdot p_1 + a \qquad p = a \cdot p_2$$
$$q = a \cdot b + a \cdot a \ .$$

First of all, we notice that both p_1 and p_2 have norm 1, due to the branches with the action constant a. Moreover, they are both uniformly determinate. In fact, p_1 can perform only one transition to process b, which has norm 1 and it is clearly order-insensitive bisimilar to itself. Similarly, the only available transition for p_2 is $p_2 \xrightarrow{a} p_1$, which, as previously noticed, has norm 1. We remark that the a action constants in p_1 and p_2 do not trigger any transition for the two processes, but they cause the predicates $p_1 \xrightarrow{a} \surd\!\!\surd$ and $p_2 \xrightarrow{a} \surd\!\!\surd$ to hold.

As process p can perform only one a-move to p_2, we can directly infer that p is uniformly determinate. Notice that process p does not have norm 1, but such a constraint has to be satisfied only by its derivatives. Moreover, from our observations on p_1 and p_2, we obtain that p is also uniformly 1-determinate and uniformly 2-determinate.

Consider now process q. We have that q is not uniformly determinate since $q \xrightarrow{a} b$ and $q \xrightarrow{a} a$ are both derivable and, clearly, $b \not\underline{\leftrightarrow}_* a$. However, q is uniformly 1-determinate, since both b and a are trivially uniformly determinate. ◀

The notion of uniform k-determinacy is preserved by order-insensitive bisimilarity.

Lemma 7. *If $p \underline{\leftrightarrow}_* q$ and p is uniformly k-determinate for all $1 \leq k < \text{depth}(p)$, then so is q.*

Proof. The proof proceeds by induction on k. Notice that $p \underline{\leftrightarrow}_* q$ implies $p \underline{\leftrightarrow} q$.

- Base case: $k = 1$. Assume, towards a contradiction, that q is not uniformly 1-determinate. This means that either $|\text{init}(q)| > 1$ or there exist q_1 and q_2 such that $q \to q_1$ and $q \to q_2$ but $q_1 \not\underline{\leftrightarrow}_\star q_2$, or $\text{norm}(q_1) \neq 1$, or $\text{norm}(q_2) \neq 1$.

  If $|\text{init}(q)| > 1$, then there are $a, b \in \mathcal{A}$ with $a \neq b$ such that $q \xrightarrow{a} q_a$ and $q \xrightarrow{b} q_b$ for some processes q_a and q_b. Since $p \underline{\leftrightarrow} q$, there must exist p_a and p_b such that $p \xrightarrow{a} p_a$ and $p \xrightarrow{b} p_b$, but this contradicts $|\text{init}(p)| = 1$.

  If $q_1 \not\underline{\leftrightarrow}_* q_2$, then $q_1 \not\underline{\leftrightarrow}_> q_2$ for some priority order $>$. Since we already know that $|\text{init}(q)| = 1$, $q \to q_1$ and $q \to q_2$ implies $q \to_> q_1$ and $q \to_> q_2$. Hence there exist processes p_1 and p_2 such that $p \to_> p_1$ and $p \to_> p_2$ with $p_1 \underline{\leftrightarrow}_> q_1$ and $p_2 \underline{\leftrightarrow}_> q_2$. However, since p is uniformly 1-determinate, we know that $p_1 \underline{\leftrightarrow}_> p_2$, so $q_1 \underline{\leftrightarrow}_> q_2$, which is a contradiction.

  If $\text{norm}(q_1) \neq 1$, then we know from $p \underline{\leftrightarrow} q$ and $q \to q_1$ that $p \to p_1$ for some process p_1 with $p_1 \underline{\leftrightarrow} q_1$. But this implies $\text{norm}(q_1) = \text{norm}(p_1) = 1$, which is a contradiction. The argument for $\text{norm}(q_2) \neq 1$ is similar.

- Inductive step: $k > 1$. Assume that q is uniformly k'-determinate for all $k' < k$. We now prove that q is also uniformly k-determinate. Assume towards a contradiction that q is not k-determinate. Then there must exist some q' such that $q \to^k q'$ and either $|\mathrm{init}(q')| > 1$ or there are q_1 and q_2 such that $q' \to q_1$ and $q' \to q_2$, but either $q_1 \not\underline{\leftrightarrow}_* q_2$, $\mathrm{norm}(q_1) \neq 1$, or $\mathrm{norm}(q_2) \neq 1$.

The cases of $|\mathrm{init}(q')| > 1$, $\mathrm{norm}(q_1) \neq 1$, and $\mathrm{norm}(q_2) \neq 1$ are essentially the same as for the base case, except that one first gets a process p' such that $p \to^k p'$, and then reasons as before on p'.

We now consider the case of $q_1 \not\underline{\leftrightarrow}_* q_2$. This implies that $q_1 \not\underline{\leftrightarrow}_> q_2$ for some priority order $>$. Since $p \underline{\leftrightarrow}_* q$, we also get $p \underline{\leftrightarrow}_> q$, and since q is uniformly k'-determinate for every $k' < k$, $q \to^k q'$ implies $q \to_>^k q'$. (Recall that all the processes reached in the sequence of k'-steps can perform only transitions with the same label). Therefore there exists a process p' such that $p \to_>^k p'$ and $p' \underline{\leftrightarrow}_> q'$. Since we already know that $|\mathrm{init}(q')| = 1$, $q' \to q_1$ and $q' \to q_2$ implies $q' \to_> q_1$ and $q' \to_> q_2$. Hence there exist p_1 and p_2 such that $p' \to_> p_1$ and $p' \to_> p_2$ as well as $p_1 \underline{\leftrightarrow}_> q_1$ and $p_2 \underline{\leftrightarrow}_> q_2$. However, since p is uniformly k-determinate, we know that $p_1 \underline{\leftrightarrow}_> p_2$, so we get $q_1 \underline{\leftrightarrow}_> q_2$, which contradicts our assumption.

□

The next proposition shows that if p and q are order-insensitive bisimilar as well as uniformly k-determinate for all k less than some n, then every sequence of n transitions that p can do can be matched by q such that p and q end up in processes that are again order-insensitive bisimilar.

Proposition 3. *Let p and q be two processes such that $p \underline{\leftrightarrow}_* q$ and there is an $n \in \mathbb{N}$ such that p and q are uniformly k-determinate for all $k < n$. Suppose that $p \to^n p'$ for some p'. Then there is a process q' such that $q \to^n q'$ and $p' \underline{\leftrightarrow}_* q'$.*

Proof. We recall that in [3] a process p is said to be determinate if $|\mathrm{init}(p)| \leq 1$ ([3] considers the language BCCSP which includes the idle process that cannot perform any action), and for all processes p_1, p_2 such that $p \to p_1$ and $p \to p_2$ it holds that $p_1 \underline{\leftrightarrow}_* p_2$. Then p is said to be determinate at depth k if all processes p' such that $p \to^k p'$ are determinate. Since our notion of uniformly k-determinacy implies that of determinacy at depth k in [3], the proof of this proposition directly follows from Lemma 18 of [3]. □

6. The special property: uniform (n, Θ)-dependency

In this section we formalize the uniform (n, Θ)-dependency property, on which our negative result is built. As previously outlined, this is based on the notion of Θ-*dependent* process from [3].

Definition 6 (Θ-dependent process, [3])**.** A process p is Θ-*dependent* if there exist priority orders $>_1$ and $>_2$ such that $\mathrm{init}_{>_1}(p) \neq \mathrm{init}_{>_2}(p)$.

Intuitively, a process is Θ-dependent if its possible behaviour depends on the choice of priority order. For example, $\Theta(a + b)$ is Θ-dependent, since we can find a priority order that only allows it to make an a-transition, and another priority order that only allows it to make a b-transition. On the other hand, $\Theta(a)$ is not Θ-dependent, since no matter what priority order we choose, it can only do a a-transition.

Moreover, we will make use of the following technical result from [3].

Lemma 8 ([3, Lemma 14])**.** *If $p \underline{\leftrightarrow}_* q$ and p is Θ-dependent, then so is q.*

Uniform (n, Θ)-dependency is an extension of Θ-dependency from [3], in that it requires first that it is possible to take a sequence of n transitions and end up in a process that is Θ-dependent, and furthermore it requires that at each step along the way, the process has a norm of 1.

Definition 7. A process p is *uniformly* (n, Θ)-*dependent* if there are processes $p_1, \ldots, p_n$ such that $p = p_0 \to p_1 \to \cdots \to p_n$, the process p_n is Θ-dependent, and for all $0 \leq k < n$ we have $\mathrm{norm}(p_k) = 1$.

The following proposition tells us that (n, Θ)-dependency is preserved by closed instantiations of sound equations whose depth is smaller than n and that satisfy some determinacy constraints.

Proposition 4. *Let σ be a closed substitution and let t and u be process terms such that $t \underline{\leftrightarrow}_* u$ and $\mathrm{init}^{\omega}(t) = \{a\}$. Assume a natural number $n \in \mathbb{N}$ such that $n > \max\{\mathrm{depth}\,(t), \mathrm{depth}\,(u)\}$ and $\sigma(t)$ is uniformly k-determinate for all $1 \leq k \leq n-1$. If $\sigma(t)$ is uniformly (n, Θ)-dependent, then so is $\sigma(u)$.*

Proof. We start by noticing that $t \underline{\leftrightarrow}_* u$ implies $\sigma(t) \underline{\leftrightarrow}_* \sigma(u)$ and thus, by Lemma7, we infer that $\sigma(u)$ is uniformly k-determinate for all $1 \leq k \leq n-1$. Next, since $\sigma(t)$ is uniformly (n, Θ)-dependent, by Definition 5 there are processes $p_0, \ldots, p_n$ such that $\sigma(t) = p_0 \to \ldots \to p_n$, $\mathrm{norm}\,(p_i) = 1$ for all $i = 0, \ldots, n-1$, and p_n is Θ-dependent. Since, moreover, we have $\mathrm{depth}\,(t) < n$, by Proposition 2 there are a process term t' and a string w such that $t \xrightarrow{s_1 \ldots s_h}_w t'$ with $|s_1 \ldots s_h| = j$ and there are a variable x, an $l \in \mathbb{N}$ and a process q such that $t' \xrightarrow{x_s} \Theta^l_{\odot}(x_d)$, $\sigma(x) \to^{n-j} q$, and $p_n \in \Theta^l_{\odot}(q)$. In particular, notice that by the uniform k-determinacy of $\sigma(t)$, for all $k = 1, \ldots, n-1$, we obtain that $|\mathrm{init}(\sigma(t'))\,| = 1$. As this set of initials is constructed with respect to the empty priority order we can also infer the following:

- $|\mathrm{init}(\sigma(x))\,| = 1$,
- $|\mathrm{init}(q_i)\,| = 1$ for all q_i, with $i = 1, \ldots, n-j-1$, such that $\sigma(x) \to q_1 \to \ldots \to q_{n-j-1} \to q$, and
- any action performed in the sequence $\sigma(x) \to {}^{n-j} q$ is locally maximal with respect to the empty priority order.

Notice that, by Lemma 6, $t' \xrightarrow{x_s} \Theta^l_{\odot}(x_d)$ is the same as $x \triangleleft_l t'$. Since, moreover, p_n is Θ-dependent, it must be the case that $|\mathcal{A}| > 1$. We can then apply Theorem 2, thus obtaining that there are a process term u' and an $m \in \mathbb{N}$ such that $u \to^j u'$ and $x \triangleleft_m u'$. Using again Lemma 6, $x \triangleleft_m u'$ is the same as $u' \xrightarrow{x_s} \Theta^m_{\odot}(x_d)$. As above, the uniform k-determinacy of $\sigma(u)$, for all $k = 1, \ldots, n-1$, guarantees that $|\mathrm{init}(\sigma(u'))\,| = 1$ and thus that $\sigma(x)$ can perform its (locally maximal) action. Thus, from $\sigma(x) \xrightarrow{a}^{n-j} q$ and $u' \xrightarrow{x_s} \Theta^m_{\odot}(x_d)$, Lemma 5 implies $\sigma(u') \xrightarrow{a}^{n-j} \Theta^m_{\odot}(q)$. Hence we can infer that there are processes $q_0, \ldots, q_n$ such that $\sigma(u) = q_0 \to \ldots \to q_n$ with $q_n \in \Theta^m_{\odot}(q)$. According to Theorem 2, we can distinguish two cases:

- Case $l > 0$. Then we can infer that $m > 0$, and thus q_n is clearly Θ-dependent.
- Case $l = 0$. Then we have that $p_n = q$ and from p_n being Θ-dependent we can infer that q is Θ-dependent. As $l = 0$ implies $m = 0$, we get that $q_n = q$ and thus q_n is Θ-dependent because q is.

To conclude, we need to show that $\mathrm{norm}\,(q_i) = 1$ for each $i = 0, \ldots, n-1$. First of all we notice that, since $\sigma(t) \underline{\leftrightarrow}_* \sigma(u)$ and $\mathrm{norm}\,(\sigma(t)) = 1$, then $\mathrm{norm}\,(\sigma(u)) = \mathrm{norm}\,(q_0) = 1$. Moreover, since $\sigma(u)$ is uniformly k-determinate for all $1 \leq k < n$, we get that $\mathrm{norm}\,(q_i) = 1$ for all $i = 1, \ldots, n-1$ is guaranteed by Definition 5. We can therefore conclude that $\sigma(u)$ is uniformly (n, Θ)-dependent. $\square$

7. Order-insensitive bisimilarity is not finitely based over $\mathbf{BPA}_{\Theta}$

This section is devoted to our main result, namely that order-insensitive bisimilarity has no finite ground-complete axiomatisation in the setting of BPA_{Θ}.

In Equation (1) in Section 3, we presented a family of infinitely many sound equations which cannot be derived from any finite axiom system which is sound modulo order-insensitive bisimilarity, which we now proceed to recall. We make use of the following processes, which are defined for each $n \in \mathbb{N}$ as

$$P_n = A_n(a) + A_n(b) + A_n(a+b),$$

where $A_0(p) = p$ and $A_n(p) = a \cdot A_{n-1}(p) + a$. Process P_n must decide at the top level whether after n steps it will end up in a, b, or $a+b$. After this choice, it can take up to n a-transitions, and at each step it can choose whether to terminate or to continue. We stress that the possibility of termination at each step is crucial, as it implies that $A_n(\cdot)$ cannot be written just with sequential composition modulo bisimilarity.

The family of equations that we consider is then

$$e_n : \quad P_n + A_n(\Theta(a+b)) \approx P_n \qquad (n \geq 0) \ .$$

We remark that the processes on the left-hand side of each equation e_n are uniformly (n, Θ)-dependent, whereas those on the right-hand side do not enjoy this property. In detail, for all $n \in \mathbb{N}$, by construction there is no occurrence of Θ in P_n nor in its derivatives, so that P_n cannot have any Θ-dependent successor. On the other hand, we have $P_n + A_n(\Theta(a+b)) \xrightarrow{a} A_{n-1}(\Theta(a+b)) \xrightarrow{a} \ldots \xrightarrow{a} A_0(\Theta(a+b)) = \Theta(a+b)$ with $\Theta(a+b)$ a Θ-dependent process and, by construction, for each $i = 1, \ldots, n$ the process $A_i(\Theta(a+b))$ has norm 1.

To proceed to the proof of Theorem 1 we need to show, in the first place, that all the equations in the family $\{e_n\}_{n\in\mathbb{N}}$ are sound. To this end we introduce the final ingredient that we need for our main result, namely the notion of *summand* of a process.

Definition 8 (Summand, [3])**.** We say that p *is a summand of* q, denoted by $p \sqsubseteq_* q$, if there exists a process r such that $p + r \leftrightarrow_* q$.

Proposition 5. *For every $n \in \mathbb{N}$, the equation $P_n + A_n(\Theta(a+b)) \approx P_n$ is sound.*

Proof. It is enough to prove that $A_n(\Theta(a+b)) \sqsubseteq_* P_n$ for all $n \in \mathbb{N}$. So, let $n \in \mathbb{N}$ and $>$ be an arbitrary priority order. Then:

- If $a > b$, then $A_n(\Theta(a+b)) \leftrightarrow_> A_n(a)$.
- If $b > a$, then $A_n(\Theta(a+b)) \leftrightarrow_> A_n(b)$.
- If a and b are unordered in $>$, then $A_n(\Theta(a+b)) \leftrightarrow_> A_n(a+b)$.

Hence, we can conclude that $A_n(\Theta(a+b)) + P_n \leftrightarrow_> P_n$ for all priority orders $>$ and naturals $n \in \mathbb{N}$, which implies $A_n(\Theta(a+b)) \sqsubseteq_* P_n$ for all $n \in \mathbb{N}$ □

Interestingly, any process p such that $p \sqsubseteq_* P_n$ must be of a specific form that inherits many of the features of P_n. In particular, such a process must be k-determinate for all k less than n.

Lemma 9. *Let p be a process and assume $p \sqsubseteq_* P_n$ for some $n \in \mathbb{N}$. Then p is uniformly k-determinate for all $1 \leq k < n$.*

Proof. We first prove that $\text{init}^k(p) = \{a\}$ for $0 \leq k < n$. We recall that since we are considering BPA_Θ with constants, and without the empty process and deadlock, for all closed process terms p it holds that $\text{init}_>(p) \neq \emptyset$ for all priority orders $>$. As $p \sqsubseteq_* P_n$, which means that $p + r \leftrightarrow_* P_n$ for some r, we have that $p + r \leftrightarrow P_n$. By Lemma 1, we infer $\text{init}^k(p+r) = \text{init}^k(P_n) = \{a\}$. Since, moreover, $\text{init}^k(p) \subseteq \text{init}^k(p+r)$, we get $\text{init}^k(p) = \{a\}$.

We now proceed by contradiction. Let $1 \leq k < n$ be the least number such that p is not uniformly k-determinate. Then there exist processes p', p_1, and p_2 such that $p \to^k p'$, $p' \to p_1$, and $p' \to p_2$, and $p_1 \not\leftrightarrow_* p_2$, or $\text{norm}\,(p_1) \neq 1$, or $\text{norm}\,(p_2) \neq 1$.

If $\text{norm}\,(p_1) \neq 1$, then $p \to^k p'$ and $p' \to p_1$, so there exists P'_n and P''_n such that $P_n \to^k P'_n$ and $P'_n \to P''_n$ with $p_1 \leftrightarrow P''_n$. But then $\text{norm}\,(p_1) = \text{norm}\,(P''_n) = 1$, which is a contradiction. A similar argument holds when $\text{norm}\,(p_2) \neq 1$.

If $p_1 \not\leftrightarrow_* p_2$, then $p_1 \not\leftrightarrow_> p_2$ for some specific priority order $>$. Notice that since $|\text{init}^i(p)| = \{a\}$ for all $0 \leq i < n$, we get that $p \to^k p'$, $p' \to p_1$, and $p' \to p_2$ implies $p \to^k_> p'$, $p' \to_> p_1$, and $p' \to_> p_2$. Since $p + r \leftrightarrow_> P_n$ for some r, there exist P'_n, P''_n, and P'''_n such that $P_n \to^k_> P'_n$, $P'_n \to_> P''_n$, and $P'_n \to_> P'''_n$ with $p_1 \leftrightarrow_> P''_n$ and $p_2 \leftrightarrow_> P''_n$. Since $\text{norm}\,(p_1) = 1 = \text{norm}\,(p_2)$, we also get $\text{norm}\,(P''_n) = 1 = \text{norm}\,(P'''_n)$. However, we see from the definition of P_n that P'_n has a unique successor with norm 1. Hence it follows that $P''_n = P'''_n$, so $p_1 \leftrightarrow_> P''_n = P'''_n \leftrightarrow_> p_2$, which contradicts $p_1 \not\leftrightarrow_> p_2$. □

We are now ready to present our main theorem, which states that for n large enough, if p and q are summands of P_n that can be proved equivalent from a finite set of sound equations, and p is uniformly (n, Θ)-dependent, then q must also be uniformly (n, Θ)-dependent.

Theorem 3. *Assume that $\mathcal{A}$ contains at least two distinct actions. Let $\mathcal{E}$ be a set of sound process equations of depth less than n, and let p and q be closed processes such that $p, q \sqsubseteq_* P_n$ and $\mathcal{E} \vdash p \approx q$. If p is uniformly (n, Θ)-dependent, then q is also uniformly (n, Θ)-dependent.*

Proof. As briefly discussed in Section 2, without loss of generality, we can disregard the symmetry rule in our inductive proof below by assuming that $u \approx t \in \mathcal{E}$ whenever $t \approx u \in \mathcal{E}$. Furthermore, we can assume that all applications of the substitution rule in derivations have a process equation from $\mathcal{E}$ as premise. This means that we only need to consider a new rule stating that all substitution instances of process equations in $\mathcal{E}$ are derivable, rather than considering the axiom rule — which states that all process equations in $\mathcal{E}$ are derivable —, and the substitution rule — which states that if a process equation is derivable, then so are all its substitution instances — separately.

We will now present the inductive argument over the number of steps in a proof of an equation $p \approx q$ from $\mathcal{E}$. We proceed by a case analysis on the last rule applied to obtain $\mathcal{E} \vdash p \approx q$.

Case 1: reflexivity and transitivity. In these cases, the proof follows immediately or by the induction hypothesis in a straightforward manner.

Case 2: variable substitution. Assume that $\mathcal{E} \vdash p \approx q$ is the result of a closed substitution instance of a process equation $t \approx u \in \mathcal{E}$, namely there exists a substitution σ such that $\sigma(t) = p$ and $\sigma(u) = q$. Since $t \approx u \in \mathcal{E}$, we have that $\mathrm{depth}(t), \mathrm{depth}(u) < n$. Moreover, from $p, q \sqsubseteq_* P_n$ it follows that $\mathrm{init}^\omega(p) = \mathrm{init}^\omega(q) = \{a\}$ and that, by Lemma 9, p and q are uniformly k-determinate for all $k \in \{1, \ldots n-1\}$. Hence by Proposition 4, we can conclude that if p is uniformly (n, Θ)-dependent, then so is q.

Case 3: congruence rule. We can distinguish three cases:

- The last rule applied in $\mathcal{E} \vdash p \approx q$ is the congruence rule for the nondeterministic choice $+$. Then there exist closed process terms p_1, p_2, q_1 and q_2 such that $p = p_1 + p_2$, $q = q_1 + q_2$, $\mathcal{E} \vdash p_1 \approx q_1$ and $\mathcal{E} \vdash p_2 \approx q_2$ by shorter proofs. Since p is uniformly (n, Θ)-dependent, there must exist a process p' such that $p \to^n p'$, where p' is Θ-dependent and every process along the transitions from p to p' has norm 1.

  We can distinguish four possible subcases, regarding how this property is derived:

  1. p_1 is uniformly (n, Θ)-dependent.
  2. p_2 is uniformly (n, Θ)-dependent.
  3. $\mathrm{norm}(p_2) = 1$, $\mathrm{norm}(p_1) \neq 1$, and there are processes $p_1^1, \ldots, p_1^n$ such that $p_1 \to p_1^1 \to \ldots \to p_1^n = p'$ and p_1^n is Θ-dependent.
  4. $\mathrm{norm}(p_1) = 1$, $\mathrm{norm}(p_2) \neq 1$, and there are processes $p_2^1, \ldots, p_2^n$ such that $p_2 \to p_2^1 \to \ldots \to p_2^n = p'$ and p_2^n is Θ-dependent.

  In cases (1) and (2) we can immediately apply the induction hypothesis obtaining, respectively, that either q_1 or q_2 is uniformly (n, Θ)-dependent, and thus that q is uniformly (n, Θ)-dependent as well.

  The cases (3) and (4) require more attention. We detail only the proof for case (3), since the one for case (4) is symmetric. Firstly, we notice that since $p, q \sqsubseteq_* P_n$ then by Lemma 9 both p and q are uniformly k-determinate for all $k \in \{1, \ldots, n-1\}$. This implies that p_1 is uniformly k-determinate for the same values of k. Moreover, as $\mathcal{E} \vdash p_1 \approx q_1$ gives $p_1 \underline{\leftrightarrow}_* q_1$ and $\mathrm{depth}(p_1) = n$, by Lemma 7 we obtain that also q_1 is uniformly k-determinate for $k \in \{1, \ldots, n-1\}$. Then, by Proposition 3 we can infer that there is a process q_1^n such that $q_1 \to^n q_1^n$ and $q_1^n \underline{\leftrightarrow}_* p_1^n$, which, by Lemma 8, implies that q_1^n is Θ-dependent. Furthermore, uniform k-determinacy ensures that all the processes $q_1^1, \ldots, q_1^{n-1}$ in the sequence $q_1 \to q_1^1 \to \ldots \to q_1^{n-1} \to q_1^n$ have norm 1. Finally, we notice that since $\mathrm{norm}(p_2) = 1$ and $\mathcal{E} \vdash p_2 \approx q_2$ implies $p_2 \underline{\leftrightarrow}_* q_2$, we can infer that $\mathrm{norm}(q_2) = 1$. By combining the properties of q_1 and q_2, we can conclude that $q = q_1 + q_2$ is uniformly (n, Θ)-dependent.

$$\frac{x \xrightarrow{a}_{>} \surd \quad \forall b > a \quad y \stackrel{b}{\not\to}_{>}}{x \lhd y \xrightarrow{a}_{>} \surd} \qquad \frac{x \xrightarrow{a}_{>} x' \quad \forall b > a \quad y \stackrel{b}{\not\to}_{>}}{x \lhd y \xrightarrow{a}_{>} x'}$$

U1	$a \lhd b \approx a$ if not $b > a$	U4	$(x \cdot y) \lhd z \approx (x \lhd z) \cdot y$
U2	$x \lhd (y \cdot z) \approx x \lhd y$	U5	$(x + y) \lhd z \approx x \lhd z + y \lhd z$
U3	$x \lhd (y + z) \approx (x \lhd y) \lhd z$	PU	$\Theta(x + y) \approx \Theta(x) \lhd y + \Theta(y) \lhd x$

Table 5: Operational semantics and some axioms of the unless operator.

- The last rule applied in $\mathcal{E} \vdash p \approx q$ is the congruence rule for the sequential composition. This means that $p = p_1 \cdot p_2$, $q = q_1 \cdot q_2$, $\mathcal{E} \vdash p_1 \approx q_1$ and $\mathcal{E} \vdash p_2 \approx q_2$ by shorter proofs. This case is vacuous, as norm $(p) \geq 2$ and therefore p cannot be uniformly (n, Θ)-dependent.

- The last rule applied in $\mathcal{E} \vdash p \approx q$ is the congruence rule for the priority operator Θ. Then there exist p' and q' such that $p = \Theta(p')$, $q = \Theta(q')$, and $\mathcal{E} \vdash p' \approx q'$ by a shorter proof. Since p is uniformly (n, Θ)-dependent, there exists a sequence of processes $p = \Theta(p') \to \Theta(p_1) \to \cdots \to \Theta(p_{n-1}) \to \Theta(p_n)$ such that norm $(\Theta(p_1)) = \ldots$ norm $(\Theta(p_{n-1})) = 1$ and $\Theta(p_n)$ is Θ-dependent. Note that, since $\Theta(p_n)$ is Θ-dependent, $|\text{init}(p_n)| \geq 2$. Moreover, from the operational rules for Θ, $p' \to p_1 \to \cdots \to p_{n-1} \to p_n$ and from the definition of norm, norm $(p_1) = \cdots =$ norm $(p_n) = 1$. From $\mathcal{E} \vdash p' \approx q'$, we derive that $p' \underline{\leftrightarrow}_* q'$. Hence, $p' \underline{\leftrightarrow} q'$ holds and therefore we get a sequence $q' \to q_1 \to \cdots \to q_n$ such that $p_n \underline{\leftrightarrow} q_n$, which implies that $|\text{init}(q_n)| \geq 2$. Thus, we infer $q = \Theta(q') \to \Theta(q_1) \to \cdots \to \Theta(q_n)$ and, since $|\text{init}(q_n)| \geq 2$, $\Theta(q_n)$ is Θ-dependent. It remains to show that norm $(\Theta(q')) =$ norm $(\Theta(q_i)) = 1$ for each $i \in \{1, \ldots, n-1\}$. As $q \sqsubseteq_* P_n$, by Lemma 9 we gather that q is uniformly k-determinate for all $1 \leq k < n$, from which it follows that norm $(\Theta(q_i)) = 1$ for all $i \in \{1, \ldots, n-1\}$. Since, moreover, $p \underline{\leftrightarrow}_* q$ and norm $(p) = 1$, we get norm $(q) = 1$ and we conclude that q is (n, Θ)-dependent.

□

As the left-hand side of the equations in (1) is uniformly (n, Θ)-dependent while the right-hand side is not, from Theorem 3 we can directly infer that for each n, the nth instance of the family of equations in (1) cannot be proved using the finite collection of all sound equations whose depth is smaller than n.

We can therefore conclude that Theorem 1 (presented in Section 3) holds.

8. On the use of auxiliary operators

In its first appearance, in [8], the priority operator was defined in terms of the simpler binary operator *unless*, denoted by $\lhd$. Informally, $\lhd$ allows us to capture the priority order among actions, as $a \lhd b$ behaves like a unless b has higher priority than a. In Table 5 we report the SOS rules defining the behavior of the unless operator, together with some valid axioms for it. In particular, axiom (PU) allows us to rewrite the behaviour of the priority operator in terms of that of unless.

***Example* 6.** Consider process $p = a \cdot (b \lhd c + c \lhd b)$. If $b > c$, then only the summand $b \lhd c$ of $(b \lhd c + c \lhd b)$ can make a transition, thus giving $p \underline{\leftrightarrow}_> a \cdot b$. Similarly, if $c > b$ then $p \underline{\leftrightarrow}_> a \cdot c$. In case b and c are incomparable with respect to $>$, then $b \lhd c + c \lhd b \underline{\leftrightarrow}_> b + c$, so that $p \underline{\leftrightarrow}_> a \cdot (b + c)$.

Consider now processes $q_1 = a \lhd (b \cdot c)$ and $q_2 = (a \cdot b) \lhd c$. The unless operator compares only the initial actions of its arguments (cf. axioms (U2) and (U4) in Table 5). Hence in q_1 the priority order between a and c plays no role in determining whether q_1 will perform the a-move or not. At the same time, if c has higher priority than a, in q_2 also the execution of b is prevented disregarding the ordering of b and c. ◀

One can prove, in a similar fashion to [8], that, provided the set of actions is finite, for a chosen priority order $>$, the bisimulation equivalence $\underline{\leftrightarrow}_>$ affords a finite axiomatisation over $\mathrm{BPA}_{\Theta,\lhd}$, namely BPA enriched with both Θ and $\lhd$. Hence a natural question that arises is whether we can regain a finite axiomatisation over $\mathrm{BPA}_{\Theta,\lhd}$ also for order-insensitive bisimilarity. We devote this section to proving that a negative answer applies and thus that the following theorem holds:

Theorem 4. *If the set of actions $\mathcal{A}$ contains at least two distinct actions, then the language $BPA_{\Theta,\lhd}$ modulo order-insensitive bisimilarity is not finitely based.*

Since the technical development of the negative result for BPA_Θ (Theorem 1) would apply in major part unchanged in the proof of Theorem 4, we actually present only an informal discussion of this result.

Consider the family of equations in (1), that we used to prove the negative result for the priority operator. One can prove, by using axioms (PU) in Table 5 and (P5) in Table 3 together with congruence closure, that

$$A_n(\Theta(a+b)) \approx A_n(a \lhd b + b \lhd a)$$

and thus that the family of equations

$$e'_n: \quad P_n + A_n(a \lhd b + b \lhd a) \approx P_n \qquad (n \geq 0) \tag{2}$$

is sound modulo order-insensitive bisimilarity. However, precisely because we are considering the order-insensitive relation, one can notice that it is not possible to eliminate the occurrences of the unless operator from the left-hand side of the equations in (2). In fact, as no priority order over actions has been chosen, it is not possible to establish the relation between actions a and b (that we recall are assumed to be distinct) and thus whether $\lhd$ will allow for their execution or not. More formally, we notice that the axiom (U1) in Table 5 is not sound modulo order-insensitive bisimilarity (with the only exception of the trivial case in which the actions in the two sides of $\lhd$ coincide). Therefore, the same reasoning applied to prove Theorem 3, and thus Theorem 1, can be adapted in a straightforward manner to obtain a proof for Theorem 4. Intuitively, we simply need to substitute the notions of Θ-dependency and uniform (n,Θ)-dependency with the corresponding notions for the unless operator.

9. Complexity of order-insensitive bisimilarity checking

In this section we investigate some algorithms, and their complexity, for checking order-insensitive bisimilarity of (loop-free) finite labelled transition systems. It is known that bisimilarity over such systems is **P**-complete [9], and, moreover, using the Paige-Tarjan algorithm [25] each $\underline{\leftrightarrow}_>$ can be checked in $O(m_t \log m_s)$, where m_t is the number of transitions, and m_s is the number of states. A naive algorithm for $\underline{\leftrightarrow}_*$ would then check $\underline{\leftrightarrow}_>$ for all the possible partial orders $>$ over $\mathcal{A}$. Assuming that $|\mathcal{A}| = k > 0$, there are $2^{k^2/4+3k/4+O(\log k)}$ possible partial orders (see [20] for the result on the number of posets over sets with k elements). Clearly, from these results we can obtain an upper bound on the complexity of $\underline{\leftrightarrow}_*$.

Theorem 5. *The problem of deciding whether two processes are order-insensitive bisimilar is in* **coNP** *and can be solved in time $2^{k^2/4+3k/4+O(\log k)} \cdot O(n^2)$ where k is the number of actions and n is the sum of the sizes of the two processes.*

Proof. Let $|p|$ denote the size of process p. We first argue that the complexity of the naive algorithm for checking whether two closed BPA_Θ terms p and q are related by order-insensitive bisimilarity is

$$2^{k^2/4+3k/4+O(\log k)} \cdot O(n^2) \ ,$$

where $n = |p| + |q|$ is the sum of the sizes of the two processes. To this end, observe that, for each irreflexive partial order $>$ over $\mathcal{A}$, the algorithm checks whether $p \,\underline{\leftrightarrow}_>\, q$ holds, which can be done by verifying that the loop-free LTSs with transition relation $\rightarrow_>$ associated with p and q are bisimilar. The latter check can be done in $O(m_t \log m_s)$ using the Paige-Tarjan algorithm. It is not hard to verify that the number m_s of states and the number m_t of transitions in the LTS associated with a closed BPA_Θ term are linear in the size of the term. Moreover such an LTS can be constructed in time $O(|p|^2)$ from a term p and a priority order $>$. So checking whether p and q are related by $\underline{\leftrightarrow}_>$ can be done in time $O(n^2 + n \log n) = O(n^2)$, where n is the sum of the sizes of p and q. It follows that the naive algorithm has complexity $2^{k^2/4+3k/4+O(\log k)} \cdot O(n^2)$.

We now argue that order-insensitive bisimilarity checking is in **coNP**. Given two terms p and q that are not order-insensitive bisimilar, one can nondeterministically guess an irreflexive partial order $>$ that separates them, generate the loop-free LTSs with transition relation $\rightarrow_>$ associated with p and q (which can be done in quadratic time), and then verify the correctness of this guess with the Paige-Tarjan algorithm that checks for bisimilarity of the LTSs. Guessing an irreflexive partial over k elements can be done by:

- Guessing an irreflexive relation in time $O(k^2)$;
- Computing its transitive closure in cubic time;
- Checking whether the resulting relation is acyclic in time that is linear in the size of the resulting directed graph.

The **coNP** bound follows from the above mentioned observations. □

Remark 1. If $\mathcal{A}$ is a singleton, the complexity bounds in Theorem 5 can be sharpened. Indeed, in that case, $\underline{\leftrightarrow}_*$ coincides with bisimilarity and checking whether two loop-free LTSs over a singleton action set are bisimilar is **P**-complete [9].

The main contributor to the complexity of the above-mentioned naive algorithm however is the number of bisimilarity checks that has to be performed. Indeed, when verifying the order-insensitive bisimilarity of two BPA_Θ terms, the only upper bound we can impose on the number of actions appearing in the terms is linear in the size of the terms in the worst case. Therefore the number of possible partial orders that have to be considered is exponential in size of the input terms. It might be possible to improve on the number of the partial orders to consider if we could exclude a priori the checking of some significant number of partial orders. For instance, one could hope that $p \underline{\leftrightarrow}_{>_0} q$ does not need to be checked if $p \underline{\leftrightarrow}_{>_1} q$ for some $>_1$ that extends $>_0$. We dedicate the remainder of this section to showing that this is impossible in general.

Assume that $\mathcal{A}$ is finite and $|\mathcal{A}| = k > 0$. Let $>_0$ be an irreflexive partial order over $\mathcal{A}$. Our goal is to construct two BPA_Θ terms p and q with the following properties:

(a) $p \not\underline{\leftrightarrow}_{>_0} q$, and

(b) $p \underline{\leftrightarrow}_> q$ for each irreflexive partial order $> \neq >_0$.

We introduce next some constructions and notation that will be useful in what follows.

First of all, for each non-empty $S \subseteq \mathcal{A}$, we define the term $v(S)$ thus:

$$v(S) = \begin{cases} a & \text{if } S = \{a\} \text{ for some } a \in \mathcal{A} \\ \sum_{a \in S} a.v(S \setminus \{a\}) & \text{otherwise.} \end{cases}$$

Intuitively, $v(S)$ describes a nondeterministic process that can perform all permutations of the actions in S.

Given an irreflexive partial order $>$ over $\mathcal{A}$, we let $p_>$ denote a closed BPA_Θ term such that $p_>$ contains no occurrences of Θ and

$$p_> \underline{\leftrightarrow}_> \Theta(v(\mathcal{A})). \tag{3}$$

***Example* 7.** Assume that $\mathcal{A} = \{a, b\}$ and let $>_0 = \emptyset$. There are only two other irreflexive partial orders over $\{a, b\}$, namely $>_1 = \{(a, b)\}$ and $>_2 = \{(b, a)\}$. Now consider the term

$$v = v(\{a, b\}) = ab + ba \ .$$

It is easy to see that

- $\Theta(v) \underline{\leftrightarrow}_{>_0} v$,
- $\Theta(v) \underline{\leftrightarrow}_{>_1} ab = p_{>_1}$, and
- $\Theta(v) \underline{\leftrightarrow}_{>_2} ba = p_{>_2}$.

Consider now processes $p = a.p_{>_1} + a.p_{>_2}$ and $q = p + a.\Theta(a.b + b.a)$. From the above, it follows immediately that $p \underline{\leftrightarrow}_{>_1} q$ and $p \underline{\leftrightarrow}_{>_2} q$. However, we have that $p \not\underline{\leftrightarrow}_{>_0} q$. Indeed, $\Theta(v) \underline{\leftrightarrow}_{>_0} v$ and thus q can do an a action and become $a.b + b.a$ while p cannot match that transition. ◀

As highlighted by the above example, the traces of the term $\Theta(v(\mathcal{A}))$ with respect to $\rightarrow_>$ are all the *linearisations* of the partial order $>$. A classic result in order theory states that a partial order is uniquely determined by its linear extension [28]. This is the key to the following lemma.

Lemma 10. *Two closed process terms $p_{>_1}$ and $p_{>_2}$ defined as in Equation* (3) *above have the same traces if and only if $>_1 = >_2$.*

Using the above lemma, we can now prove that:

Theorem 6. *Assume that $\mathcal{A}$ is finite and contains at least two distinct actions. Let $>_0$ be an irreflexive partial order over $\mathcal{A}$. Then there exist closed BPA$_\Theta$ terms p and q such that, for each irreflexive partial order $>$ over $\mathcal{A}$, $p \underline{\leftrightarrow}_> q$ if and only if $> \neq >_0$.*

Proof. We need to exhibit two closed BPA$_\Theta$ terms satisfying the above-mentioned properties (a) and (b) with respect to the chosen partial order $>_0$. To this end, we choose an action $a \in \mathcal{A}$ and define:

$$p = \sum_{> \in PO(\mathcal{A}),\ > \neq >_0} a.p_> \quad \text{and} \quad q = p + a\Theta(v(\mathcal{A})) \tag{4}$$

where $PO(\mathcal{A})$ denotes the set of all irreflexive partial orders on $\mathcal{A}$.

- p AND q SATISFY PROPERTY (b).

  We need to show that $p \underline{\leftrightarrow}_> q$ for each $> \in PO(\mathcal{A})$ such that $> \neq >_0$. This follows by construction. In fact, for each $> \neq >_0$, both processes contain a summand bisimilar to the closed term $a.p_>$ and moreover $a.\Theta(v(\mathcal{A})) \underline{\leftrightarrow}_> a.p_>$.

- p AND q SATISFY PROPERTY (a).

  We need to show that $p \not\underline{\leftrightarrow}_{>_0} q$. To see this, observe that $q \xrightarrow{a}_{>_0} \Theta(v(\mathcal{A})) \underline{\leftrightarrow}_{>_0} p_{>_0}$. On the other hand, if $p \xrightarrow{a}_{>_0} p'$ then $p' = p_> \underline{\leftrightarrow}_> \Theta(v(\mathcal{A}))$ for some partial order $> \neq >_0$. By Lemma 10, $p_>$ does not have the same traces as $p_{>_0}$, and thus $p_{>_0} \not\underline{\leftrightarrow}_{>_0} p_>$. This means that p cannot match the transition $q \xrightarrow{a}_{>_0} \Theta(v(\mathcal{A}))$ up to $\underline{\leftrightarrow}_{>_0}$ and thus $p \not\underline{\leftrightarrow}_{>_0} q$.

□

10. Conclusions

In this work we have studied the finite axiomatisability of the equational theory of order-insensitive bisimilarity over the language BPA enriched with the priority operator Θ. As previous similar work suggested, also in this setting, the collection of sound, closed equations is not finitely based in the presence of at least two actions, despite the fact that the sequential composition operator allows one to write more complex axioms than action prefixing. We proved this negative result using an infinite family of closed equations suggested in [3] and showing that no set of sound equations of bounded depth can derive them all.

Finding an infinite (ground-)complete axiomatisation of order-insensitive bisimilarity is a natural avenue for future research. It would also be interesting to see whether we can obtain a lower bound on the complexity of order-insensitive bisimilarity checking. Above we discussed various upper bounds for its complexity that all suggest some type of computational hardness and since we have that the problem is in **coNP** it would be a natural follow-up to prove **coNP**-hardness. At the time of writing, this hardness result is not obvious to us.

Acknowledgements We thank the reviewers for their constructive feedback and careful reading of our paper. The work reported in this paper has been supported by the project '*Open Problems in the Equational Logic of Processes*' (OPEL) of the Icelandic Research Fund (grant nr. 196050-051).

References

[1] Aceto, L., Anastasiadi, E., Castiglioni, V., Ingólfsdóttir, A., & Pedersen, M. R. (2019). On the axiomatizability of priority III: the return of sequential composition. In *Proceedings of ICTCS 2019* (pp. 145–157). volume 2504 of *CEUR Workshop Proceedings*. URL: `http://ceur-ws.org/Vol-2504/paper18.pdf`.

[2] Aceto, L., Chen, T., Fokkink, W., & Ingólfsdóttir, A. (2006). On the axiomatizability of priority. In *Proceedings of ICALP 2006, (2)* (pp. 480–491). volume 4052 of *Lecture Notes in Computer Science*. doi:`10.1007/11787006_41`.

[3] Aceto, L., Chen, T., Ingólfsdóttir, A., Luttik, B., & van de Pol, J. (2011). On the axiomatizability of priority II. *Theoretical Computer Science*, *412*, 3035–3044. doi:`10.1016/j.tcs.2011.02.033`.

[4] Aceto, L., Fokkink, W., Ingólfsdóttir, A., & Luttik, B. (2005). Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday* (pp. 338–367). volume 3838 of *Lecture Notes in Computer Science*. URL: `https://doi.org/10.1007/11601548_18`.

[5] Aceto, L., Fokkink, W., Ingólfsdóttir, A., & Nain, S. (2006). Bisimilarity is not finitely based over BPA with interrupt. *Theoretical Computer Science*, *366*, 60–81. doi:`10.1016/j.tcs.2006.07.003`.

[6] Aceto, L., Fokkink, W. J., & Verhoef, C. (2001). Structural operational semantics. In *Handbook of Process Algebra* (pp. 197–292). Elsevier.

[7] Baeten, J., Basten, T., & Reniers, M. (2010). *Process algebra : equational theories of communicating processes*. Cambridge tracts in theoretical computer science. United Kingdom: Cambridge University Press. doi:`10.1017/CBO9781139195003`.

[8] Baeten, J. C., Bergstra, J. A., & Klop, J. W. (1986). Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, *IX*, 127–168.

[9] Balcázar, J. L., Gabarró, J., & Santha, M. (1992). Deciding bisimilarity is P-complete. *Formal Aspects of Computing*, *4*, 638–648.

[10] Bergstra, J. (1985). *Put and get, primitives for synchronous unreliable message passing*. Logic Group Preprint Series Nr. 3 CIF, State University of Utrecht.

[11] Bergstra, J. A., & Klop, J. W. (1984). Process algebra for synchronous communication. *Information and Control*, *60*, 109–137. doi:`10.1016/S0019-9958(84)80025-X`.

[12] Bergstra, J. A., Ponse, A., & Smolka, S. A. (Eds.) (2001). *Handbook of Process Algebra*. North-Holland / Elsevier. URL: `https://doi.org/10.1016/b978-0-444-82830-9.x5017-6`. doi:`10.1016/b978-0-444-82830-9.x5017-6`.

[13] Bloom, B., Istrail, S., & Meyer, A. R. (1995). Bisimulation can't be traced. *Journal of ACM*, *42*, 232–268.

[14] Bol, R. N., & Groote, J. F. (1996). The meaning of negative premises in transition system specifications. *J. ACM*, *43*, 863–914. URL: `http://doi.acm.org/10.1145/234752.234756`. doi:`10.1145/234752.234756`.

[15] Cleaveland, R., Lüttgen, G., & Natarajan, V. (2001). Priority in process algebra. In J. Bergstra, A. Ponse, & S. Smolka (Eds.), *Handbook of Process Algebra* (pp. 711 – 765). Amsterdam: Elsevier Science. URL: `https://doi.org/10.1016/B978-044482830-9/50030-8`.

[16] van Glabbeek, R. J. (1996). The meaning of negative premises in transition system specifications II. In *Proceedings of ICALP'96* Lecture Notes in Computer Science (pp. 502–513). doi:`10.1007/3-540-61440-0_154`.

[17] Groote, J. F. (1993). Transition system specifications with negative premises. *Theoret. Comput. Sci.*, *118*, 263–299.

[18] Hoare, C. A. R. (1985). *Communicating Sequential Processes*. Prentice-Hall.

[19] Keller, R. M. (1976). Formal verification of parallel programs. *Commun. ACM*, *19*, 371–384. URL: `https://doi.org/10.1145/360248.360251`. doi:`10.1145/360248.360251`.

[20] Kleitman, D. J., & Rothschild, B. L. (1975). Asymptotic enumeration of partial orders on a finite set. *Transaction of American Mathematical Society*, *205*, 205–220. doi:`10.2307/1997200`.

[21] Milner, R. (1989). *Communication and concurrency*. PHI Series in computer science. Prentice Hall.

[22] Moller, F. (1989). *Axioms for Concurrency*. Ph.D. thesis Department of Computer Science, University of Edinburgh. Report CST-59-89. Also published as ECS-LFCS-89-84.

[23] Moller, F. (1990). The importance of the left merge operator in process algebras. In *Proceedings of ICALP '90* (pp. 752–764). volume 443 of *Lecture Notes in Computer Science*. doi:`10.1007/BFb0032072`.

[24] Moller, F. (1990). The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings of LICS '90* (pp. 142–153). doi:`10.1109/LICS.1990.113741`.

[25] Paige, R., & Tarjan, R. E. (1987). Three partition refinement algorithms. *SIAM J. Comput.*, *16*, 973–989. doi:`10.1137/0216062`.

[26] Park, D. M. R. (1981). Concurrency and automata on infinite sequences. In *Proceedings of GI-Conference* (pp. 167–183). volume 104 of *Lecture Notes in Computer Science*. URL: `https://doi.org/10.1007/BFb0017309`. doi:`10.1007/BFb0017309`.

[27] Plotkin, G. D. (1981). *A structural approach to operational semantics*. Report DAIMI FN-19 Aarhus University.

[28] Szpilrajn, E. (1930). Sur l'extension de l'ordre partiel. *Fundamenta Matematicae*, *16*, 386–389. URL: http://eudml.org/doc/212499.

[29] Vrancken, J. L. M. (1997). The algebra of communicating processes with empty process. *Theoretical Computer Science*, *177*, 287–328. doi:10.1016/S0304-3975(96)00250-2.

Appendix A. Proofs of results in Section 4

Appendix A.1. Proof of Lemma 3

Proof of Lemma 3.

1. We proceed by induction over the derivation of the predicate $t \xrightarrow{x}_{>} \surd\!\!\!\surd$.
   - Base case: $t = x$ and $t \xrightarrow{x}_{>} \surd\!\!\!\surd$ is derived by rule (a_2) in Table 4. Hence $\sigma(t) \xrightarrow{a}_{>} \surd\!\!\!\surd$ directly follows by $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$.
   - Inductive step: $t = t_1 + t_2$ and $t \xrightarrow{x}_{>} \surd\!\!\!\surd$ is derived by either rule (a_8) in Table 4, and thus by $t_1 \xrightarrow{x}_{>} \surd\!\!\!\surd$, or its symmetric version on t_2. Assume, without loss of generality, that rule (a_8) in Table 4 was applied. Then by induction $t_1 \xrightarrow{x}_{>} \surd\!\!\!\surd$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply $\sigma(t_1) \xrightarrow{a}_{>} \surd\!\!\!\surd$. Hence, the premise of rule (r_4) in Table 1 is satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \surd\!\!\!\surd$.
   - Inductive step: $t = \Theta(u)$ and $t \xrightarrow{x}_{>} \surd\!\!\!\surd$ is derived by rule (a_{11}) in Table 4, and thus we have that $u \xrightarrow{x}_{>} \surd\!\!\!\surd$. By induction $u \xrightarrow{x}_{>} \surd\!\!\!\surd$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply $\sigma(u) \xrightarrow{a}_{>} \surd\!\!\!\surd$. Since, per assumption, action a has maximal priority with respect to $>$, the premises of rule (r_8) in Table 1 are satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \surd\!\!\!\surd$.
2. We proceed by induction over the derivation of the auxiliary transition $t \xrightarrow{x}_{>} t'$.
   - Base case: $t = t_1 \cdot t_2$ and $t \xrightarrow{x}_{>} t'$ is derived by rule (a_5) in Table 4, namely $t_1 \xrightarrow{x}_{>} \surd\!\!\!\surd$ and $t' = t_2$. By Lemma 3.1 we have that $t_1 \xrightarrow{x}_{>} \surd\!\!\!\surd$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply that $\sigma(t_1) \xrightarrow{a}_{>} \surd\!\!\!\surd$. Hence, the premise of rule (r_2) in Table 1 is satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma(t_2)$.
   - Inductive step: $t = t_1 \cdot t_2$ and $t \xrightarrow{x}_{>} t'$ is derived by rule (a_4) in Table 4, namely $t_1 \xrightarrow{x}_{>} t_1'$ and $t' = t_1' \cdot t_2$. By induction we have that $t_1 \xrightarrow{x}_{>} t_1'$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply that $\sigma(t_1) \xrightarrow{a}_{>} \sigma(t_1')$. Hence, the premise of rule (r_3) in Table 1 is satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma(t_1' \cdot t_2)$.
   - Inductive step: $t = t_1 + t_2$ and $t \xrightarrow{x}_{>} t'$ is derived either by rule (a_7) in Table 4, namely $t_1 \xrightarrow{x}_{>} t_1'$ and $t' = t_1'$, or by its symmetric version for t_2. Assume, without loss of generality, that rule (a_7) was applied. By induction we have that $t_1 \xrightarrow{x}_{>} t_1'$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply that $\sigma(t_1) \xrightarrow{a}_{>} \sigma(t_1')$. Hence, the premise of rule (r_6) in Table 1 is satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma(t_1')$.
   - Inductive step: $t = \Theta(u)$ and $t \xrightarrow{x}_{>} t'$ is derived by rule (a_{10}) in Table 4, namely $t_1 \xrightarrow{x}_{>} t_1'$ and $t' = \Theta(t_1')$. By induction we have that $t_1 \xrightarrow{x}_{>} t_1'$ and $\sigma(x) \xrightarrow{a}_{>} \surd\!\!\!\surd$ imply that $\sigma(t_1) \xrightarrow{a}_{>} \sigma(t_1')$. Since by the hypothesis action a has maximal priority with respect to $>$, the premise of rule (r_9) in Table 1 is satisfied and we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma(\Theta(t_1'))$.
3. We proceed by induction over the derivation of the auxiliary transition $t \xrightarrow{x_s}_{>} c$.
   - Base case: $t = x$ and $t \xrightarrow{x_s}_{>} c$ is derived by rule (a_1) in Table 4, namely $c = x_d$. Hence the proof follows directly by $\sigma(x) \xrightarrow{a}_{>} p$.
   - Inductive step: $t = t_1 \cdot t_2$ and $t \xrightarrow{x_s}_{>} c$ is derived by rule (a_3) in Table 4, namely $t_1 \xrightarrow{x_s}_{>} c'$ and $c = c' \cdot t_2$. By induction we have that $t_1 \xrightarrow{x_s}_{>} c'$ and $\sigma(x) \xrightarrow{a}_{>} p$ imply $\sigma(t_1) \xrightarrow{a}_{>} p'$ for $p' = \sigma[x_d \mapsto p](c')$. Hence, by rule (r_3) in Table 1 we can infer that $\sigma(t) \xrightarrow{a}_{>} p' \cdot \sigma(t_2)$, with $p' \cdot \sigma(t_2) = \sigma[x_d \mapsto p](c' \cdot t_2)$.
   - Inductive step: $t = t_1 + t_2$ and $t \xrightarrow{x_s}_{>} c$ is derived either by rule (a_6) in Table 4, namely $t_1 \xrightarrow{x_s}_{>} c$, or by its symmetric version for t_2. Assume, without loss of generality, that (a_6) was applied. By induction we have that $t_1 \xrightarrow{x_s}_{>} c$ and $\sigma(x) \xrightarrow{a}_{>} p$ imply $\sigma(t_1) \xrightarrow{a}_{>} \sigma[x_d \mapsto p](c)$. Hence, by rule (r_6) in Table 1 we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma[x_d \mapsto p](c)$.

- Inductive step: $t = \Theta(u)$ and $t \xrightarrow{x_s}_{>} \Theta(c)$ is derived by rule (a_9) in Table 4, namely $u \xrightarrow{x_s}_{>} c$. By induction we have that $u \xrightarrow{x_s}_{>} c$ and $\sigma(x) \xrightarrow{a}_{>} p$ imply $\sigma(u) \xrightarrow{a}_{>} \sigma[x_d \mapsto p](c)$. Since by the hypothesis action a has maximal priority with respect to $>$, by rule (r_9) in Table 1 we can infer that $\sigma(t) \xrightarrow{a}_{>} \sigma[x_d \mapsto p](\Theta(c))$.

□

Appendix A.2. Proof of Lemma 4

Proof of Lemma 4. We proceed by structural induction on c.

- Base case $c = t$: since c does not contain an occurrence of x_d, the lemma is vacuously true.
- Base case $c = x_d$: clearly, $\sigma[x_d \mapsto p](c) = p \xrightarrow{a}_{>} p' = \sigma[x_d \mapsto p'](c)$.
- Inductive step $c = c' \cdot t$: by induction over c' we obtain $\sigma[x_d \mapsto p](c') \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](c')$. An application of rule (r_3) in Table 1 therefore gives

$$\sigma[x_d \mapsto p](c) = \sigma[x_d \mapsto p](c') \cdot \sigma(t) \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](c') \cdot \sigma(t) = \sigma[x_d \mapsto p'](c).$$

- Inductive step $c = \Theta(c')$: by induction over c' we have $\sigma[x_d \mapsto p](c') \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](c')$. Since moreover a is maximal with respect to $>$, by applying rule (r_9) in Table 1 we obtain

$$\sigma[x_d \mapsto p](c) = \sigma[x_d \mapsto p](\Theta(c')) \xrightarrow{a}_{>} \sigma[x_d \mapsto p'](\Theta(c')) = \sigma[x_d \mapsto p'](c).$$

□

Appendix A.3. Proof of Lemma 5

Proof of Lemma 5. First of all, we notice that since $t \xrightarrow{x_s}_{>} c$, then c must contain an occurrence of x_d.

We proceed by induction over the derivation of the auxiliary transition $t \xrightarrow{x_s}_{>} c$, and for each case, we prove the statement by proceeding by induction over n. However, in each case, the base case of $n = 1$ is given by Lemma 3.3 and it is therefore omitted. Furthermore, we remark that $\sigma(x) \rightarrow^{n}_{>} p$ can be equivalently rewritten as $\sigma(x) \rightarrow^{n-1}_{>} p' \rightarrow_{>} p$ for some process p'.

- Base case: $t = x$ and $t \xrightarrow{x_s}_{>} c$ is derived by applying rule (a_1) in Table 4, so that $c = x_d$. By the induction hypothesis over $n-1$ we get

$$\sigma(t) = \sigma(x) \rightarrow^{n-1}_{>} \sigma[x_d \mapsto p'](x_d) = p'.$$

Since, moreover, $p' \rightarrow_{>} p = \sigma[x_d \mapsto p](c)$ we conclude that $\sigma(t) \rightarrow^{n}_{>} \sigma[x_d \mapsto p](c)$.

- Inductive step: $t = t_1 \cdot t_2$ and $t \xrightarrow{x_s}_{>} c$ is derived by applying rule (a_3) in Table 4, so that $t_1 \xrightarrow{x_s}_{>} c'$, and $c = c' \cdot t_2$. By induction over the derivation of $t_1 \xrightarrow{x_s}_{>} c'$ and $n-1$, we get $\sigma(t_1) \rightarrow^{n-1}_{>} \sigma[x_d \mapsto p'](c')$, which, by rule (r_3) in Table 1, gives

$$\sigma(t) = \sigma(t_1) \cdot \sigma(t_2) \rightarrow^{n-1}_{>} \sigma[x_d \mapsto p'](c') \cdot \sigma(t_2) = \sigma[x_d \mapsto p'](c).$$

Since $p' \rightarrow_{>} p$, Lemma 4 gives $\sigma[x_d \mapsto p'](c) \rightarrow_{>} \sigma[x_d \mapsto p](c)$. We can therefore conclude that $\sigma(t) \rightarrow^{n}_{>} \sigma[x_d \mapsto p](c)$.

- Inductive step: $t = t_1 + t_2$ and $t \xrightarrow{x_s}_{>} c$ is derived by applying rule (a_6) in Table 4, so that $t_1 \xrightarrow{x_s}_{>} c$. By induction over the derivation of $t_1 \xrightarrow{x_s}_{>} c$ and $n-1$, we get $\sigma(t_1) \rightarrow^{n-1}_{>} \sigma[x_d \mapsto p'](c)$. Then, by applying rule (r_6) in Table 1 and Lemma 4 we obtain

$$\sigma(t) \rightarrow^{n-1}_{>} \sigma[x_d \mapsto p'](c) \rightarrow_{>} \sigma[x_d \mapsto p](c).$$

A similar argument, using rule (r_7), in place of rule (r_6), allows us to prove the symmetric case of the auxiliary transition triggered by t_2.

- Inductive step: $t = \Theta(t')$ and $t \xrightarrow{x_s}_> c$ is derived by applying rule (a_9) in Table 4, so that $t' \xrightarrow{x_s}_> c'$ and $c = \Theta(c')$. By induction over the derivation of $t' \xrightarrow{x_s}_> c'$ and $n-1$, we infer that $\sigma(t') \to_>^{n-1} \sigma[x_d \mapsto p'](c')$. Hence, by applying rule (r_9) in Table 1 and Lemma 4, we get

$$\sigma(t) \to_>^{n-1} \sigma[x_d \mapsto p'](\Theta(c')) = \sigma[x_d \mapsto p'](c) \to_> \sigma[x_d \mapsto p](c).$$

□

Appendix A.4. Proof of Lemma 6

Proof of Lemma 6. We prove the two implications separately. We recall that $x_d \in \Theta^0_\odot(x_d)$. Moreover, we notice that if $t = a$, then there is no variable x such that either $x \lhd_l t$, or transition $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$, can be inferred for any $l \in \mathbb{N}$.

($\implies$) We proceed by structural induction on t in $x \lhd_l t$.

- Base case $t = x$. In this case we have $x \lhd_0 x$ and hence an application of rule (a_1) in Table 4 gives $t \xrightarrow{x_s}_> x_d \in \Theta^0_\odot(x_d)$.

- Inductive step $t = t_1 + t_2$. In this case $x \lhd_l t$ may be due either to $x \lhd_l t_1$ or to $x \lhd_l t_2$. If $x \lhd_l t_1$, then by induction over t_1 we get $t_1 \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$, so rule (a_6) in Table 4 gives $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$. If $x \lhd_l t_2$, we get the same by result by the symmetric version of rule (a_6).

- Inductive step $t = t_1 \cdot t_2$. Then it must be the case that $x \lhd_l t_1$, so by induction over t_1 we get $t_1 \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$. As $u \cdot t_2 \in \Theta^l_\odot(x_d)$ for all $u \in \Theta^l_\odot(x_d)$, an application of rule (a_3) in Table 4 then gives $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$, which is still of the correct form.

- Inductive step $t = \Theta(t')$. In this case $x \lhd_l t$ is due to $x \lhd_{l-1} t'$. By induction over t' we get $t' \xrightarrow{x_s}_> \Theta^{l-1}_\odot(x_d)$. Hence, since $\Theta(u) \in \Theta^l_\odot(x_d)$ for all $u \in \Theta^{l-1}_\odot(x_d)$, by applying rule (a_9) in Table 4 we obtain $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$.

($\impliedby$) The proof is by induction on the derivation of the auxiliary transition $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$.

- Base case: $t = x$ and $t \xrightarrow{x_s}_> x_d \in \Theta^0_\odot(x_d)$ is derived by applying rule (a_1) in Table 4. We can immediately infer that $l = 0$ and $x \lhd_0 t$.

- Inductive step: $t = t_1 \cdot t_2$ and $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$ is derived by applying rule (a_3) in Table 4, so that $t_1 \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$. By induction over the derivation of the auxiliary transition from t_1, we get $x \lhd_l t_1$, which implies $x \lhd_l t_1 \cdot t_2 = t$.

- Inductive step: $t = t_1 + t_2$ and $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$ is derived by applying rule (a_6) in Table 4, so that $t_1 \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$. Induction over the derivation of the auxiliary transition from t_1 then gives $x \lhd_l t_1$, which implies $x \lhd_l t_1 + t_2 = t$. The same argument holds for the symmetric version of rule (a_6).

- Inductive step: $t = \Theta(t')$ and $t \xrightarrow{x_s}_> \Theta^l_\odot(x_d)$ is derived by applying rule (a_9) in Table 4, so that $t' \xrightarrow{x_s}_> \Theta^{l-1}_\odot(x_d)$. By induction over the derivation of the auxiliary transition from t', we get $x \lhd_{l-1} t'$, which implies $x \lhd_l \Theta(t') = t$.

□